

DANIEL PIOLI TORRES

**Impacto do Gerenciamento Ativo de Filas em Redes IP
Transportando Fluxos de Vídeo H.264/MPEG-4 AVC**

CURITIBA

2009

TERMO DE APROVAÇÃO

DANIEL PIOLI TORRES

IMPACTO DO GERENCIAMENTO ATIVO DE FILAS EM REDES IP
TRANSPORTANDO FLUXOS DE VÍDEO H.264/MPEG-4 AVC

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador:



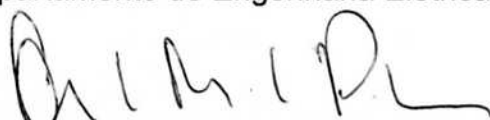
Prof. Dr. Evelio Martín García Fernández
Departamento de Engenharia Elétrica, UFPR



Profa. Dra. Keiko Verônica Ono Fonseca
Universidade Tecnológica Federal do Paraná (UTFPR)



Prof. Dr. Eduardo Parente Ribeiro
Departamento de Engenharia Elétrica, UFPR



Prof. Dr. Carlos Marcelo Pedroso
Departamento de Engenharia Elétrica, UFPR

Curitiba, 27 de março de 2009

DANIEL PIOLI TORRES

Impacto do Gerenciamento Ativo de Filas em Redes IP Transportando Fluxos de Vídeo H.264/MPEG-4 AVC

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre no Programa de Pós-
Graduação em Engenharia Elétrica, Setor de Tecnologia,
Universidade Federal do Paraná - UFPR

Orientador: Prof. Dr. Evelio Martín García Fernández

CURITIBA

2009

Agradecimentos

À minha família, pelo apoio e incentivo desde o início.

Ao Prof. Evelio, por acreditar no meu empenho e aceitar este desafio.

Ao Prof. Vitalio, por generosamente ceder os materiais que serviram de embrião para esta pesquisa.

Ao Cezar Oliveira, pela ajuda à distância para obtenção dos resultados.

Ao colega Maxwell Freitas, por me levar ao encontro do meu futuro orientador.

Aos professores e colegas do PPGE, pela valiosa troca de conhecimentos.

À minha querida esposa, pela paciência e compreensão nos momentos de ausência.

Sumário

Lista de Acrônimos	4
Lista de Figuras	5
Resumo	7
Abstract	8
1. INTRODUÇÃO	9
2. CONCEITOS FUNDAMENTAIS E REVISÃO BIBLIOGRÁFICA.....	12
2.1 A Internet como Meio de Transmissão	12
2.2 Exemplo de operação do IP.....	12
2.3 Característica <i>Best Effort</i> da Internet.....	13
2.4 Paradigma de Tráfego TCP	14
2.5 Transmissão de Dados Através da Combinação UDP/IP.....	14
2.6 Característica Inelástica dos Dados em Tempo-Real.....	16
2.7 Técnicas de Controle de Congestionamento	18
2.8 RED e ARED	18
2.9 Transmissão de Vídeo Comprimido em Redes IP	20
2.10 Taxa de Bit Constante versus Taxa de Bit Variável.....	22
2.11 Principais Aplicações da Transmissão de Vídeo na Internet.....	23
2.12 Conceitos Básicos da Codificação de Vídeo	23
2.13 O Padrão H.264 / MPEG-4 <i>Advanced Video Coding</i> (H.264/AVC).....	25
2.14 As Camadas VCL e NAL do Padrão H.264/AVC	27
2.15 Detalhamento da camada NAL	27
2.16 Detalhamento da camada VCL	28
2.17 Variabilidade de Tráfego	30
3. APRESENTAÇÃO DA METODOLOGIA	31
3.1 Topologia e Configuração da Rede Simulada	31
3.2 Fluxos de Vídeo Simulados	33
3.3 A Estrutura “Evalvid”	35
4. RESULTADOS.....	37
4.1 Impacto da Técnica ARED em Fluxos VoIP	37
4.2 Perdas de Informação de Vídeo	40
4.2.1 Técnica Droptail.....	40
4.2.2 Técnica ARED	41
4.2.3 Comparação entre as Técnicas <i>Droptail</i> e ARED	45
4.3 Atrasos Médios, Desvio Padrão dos Atrasos e Jitter.....	52
4.4 <i>Goodput</i> dos Fluxos TCP Concorrentes	56
4.5 PSNR (<i>Peak Signal-to-Noise Ratio</i>).....	58
4.5.1 Vídeo “News 200”	58
4.5.2 Vídeo “News 1024”	59
4.5.3 Vídeo “Coastguard 1024”	61
5. DISCUSSÃO DOS RESULTADOS	62
5.1 Revisão dos Resultados	62
5.2 Discussão dos Resultados	65
6. CONCLUSÕES E TRABALHOS FUTUROS	66
7. BIBLIOGRAFIA	68
Anexo I - Script base das simulações no NS2.....	70
Anexo II - Parâmetros de Codificação dos Vídeos.....	73
Anexo III - Artigo publicado no CITTEL XIV CCIA 08.....	81

Lista de Acrônimos

AQM	Active Queue Management	Gerenciamento ativo de fila
ARED	Adaptive Random Early Detection	Detecção adiantada aleatória na sua versão adaptativa
ATM	Asynchronous Transfer Mode	Modo de transferência assíncrono
AVC	Advanced Video Coding	Codificação de vídeo avançada
AVQ	Adaptive Virtual Queue	Fila virtual adaptativa
CIF	Common Intermediate Format	Formato intermediário comum
DCT	Discrete Cosine Transform	Transformada discreta do cosseno
DiffServ	Differentiated Services	Serviços diferenciados
EWMA	Exponential Weighted Moving Average	Média ponderada exponencial móvel
FEC	Forward Error Correction	Correção antecipada de erros
FTP	File Transfer Protocol	Protocolo de transferência de arquivos
GOP	Group of Pictures	Grupo de imagens
GPON	Gigabit-capable Passive Optical Network	Rede Óptica Passiva com capacidade de Gigabits por segundo
HD	High Definition	Alta definição
HTTP	Hyper Text Transfer Protocol	Protocolo de transferência de hipertexto
IEC	International Electrotechnical Commission	Comissão Eletrotécnica Internacional
IP	Internet Protocol	Protocolo da Internet
IPTV	Internet Protocol Television	Televisão sobre o Protocolo da Internet
ITU-T	Telecommunication Standardization Sector of the International Telecommunication Union	Setor de padronização de telecomunicações da União Internacional de Telecomunicações
LAN	Local Area Network	Rede local
MOS	Mean Opinion Score	Nota de opinião média
MPEG	Moving Picture Experts Group	Grupo de especialistas em imagens em movimento
NAL	Network Abstraction Layer	Camada de abstração de rede
NS2	Network Simulator version 2	Simulador de Redes versão 2
OSI	Open System Interconnection	Interconexão de sistemas abertos
PC	Personal Computer	Computador pessoal
PI	Proportional-Integral	Proporcional-Integral
PSNR	Peak Signal-to-Noise Ratio	Razão sinal-ruído de pico
QCIF	Quarter CIF	Um quarto do formato CIF
QoS	Quality of Service	Qualidade de serviço
RED	Random Early Detection	Detecção adiantada aleatória
REM	Random Exponential Marking	Marcação exponencial aleatória
RSVP	Resource Reservation Protocol	Protocolo de reserva de recursos
RTCP	Real-Time Control Protocol	Protocolo de controle de tempo real
RTP	Real-Time Transport Protocol	Protocolo de transporte em tempo real
SMTP	Simple Mail Transfer Protocol	Protocolo simples de transferência de correio
SNMP	Simple Network Management Protocol	Protocolo simples de gerência de rede
TCP	Transmission Control Protocol	Protocolo de controle de transmissão
UDP	User Datagram Protocol	Protocolo de datagrama de usuário
VBR	Variable Bit Rate	Taxa de bits variável
VCL	Video Coding Layer	Camada de codificação de vídeo
VLC	Variable-Length Coding	Codificação de comprimento variável
VoIP	Voice over Internet Protocol	Voz sobre o Protocolo da Internet
WAN	Wide Area Network	Rede de longa distância
xDSL	Digital Subscriber Line	Linha Digital de Assinante

Lista de Figuras

Figura 1 - Topologia típica de um sistema de IPTV [3].	10
Figura 2 - Topologia de um sistema de Internet TV.	11
Figura 3 - Hierarquia de protocolos do modelo OSI.	12
Figura 4 - Exemplo de operação do protocolo IP.	13
Figura 5 - Exemplo de encapsulamento RTP sobre UDP sobre IP.	15
Figura 6 - Exemplo de arquitetura de transmissão de fluxo de vídeo em tempo-real.	16
Figura 7 - Exemplo do efeito de bordas quadriculadas causado pela compressão.	21
Figura 8 - Exemplo do efeito causado por perdas de pacotes na transmissão.	21
Figura 9 - Campos envolvidos nos dois formatos básicos de vídeo.	24
Figura 10 - Diagrama de blocos do codificador H.264/AVC.	26
Figura 11 - Diagrama de blocos do decodificador H.264/AVC.	27
Figura 12 - Macrobloco e amostras Y, Cb e Cr no formato 4:2:0.	28
Figura 13 - Imagem de uma seqüência de vídeo dividida em macroblocos e fatias.	29
Figura 14 - Topologia da rede simulada.	32
Figura 15 – Tempos de início e fim dos fluxos simulados.	32
Figura 16 - Quadros da seqüência de vídeo “News”.	33
Figura 17 - Quadros da seqüência de vídeo “Coastguard”.	33
Figura 18 - Estrutura usada na codificação, simulação e avaliação dos fluxos de vídeo.	36
Figura 19 - Atraso de rede dos pacotes de voz sob a técnica <i>droptail</i> obtido por Vitalio <i>et al.</i>	38
Figura 20 - Atraso de rede dos pacotes de voz sob a técnica ARED obtido por Vitalio <i>et al.</i>	38
Figura 21 - MOS para qualidade conversacional, comparando as técnicas <i>droptail</i> e ARED, obtido por Vitalio <i>et al.</i> [4] (reproduzida com autorização dos autores).	39
Figura 22 - Vídeo “News” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica <i>droptail</i> nos roteadores.	40
Figura 23 - Vídeo “Coastguard” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica <i>droptail</i> nos roteadores.	41
Figura 24 - Vídeo “News 200” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.	42
Figura 25 - Vídeo “News 1024” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.	43
Figura 26 - Vídeo “Coastguard 200” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.	44
Figura 27 - Vídeo “Coastguard 1024” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.	44
Figura 28 - Vídeo “News 200” - Porcentagem de pacotes perdidos na comparação entre as técnicas <i>droptail</i> e ARED.	45
Figura 29 - Vídeo “News 1024” - Porcentagem de pacotes perdidos na comparação entre as técnicas <i>droptail</i> e ARED.	46
Figura 30 - Vídeo “Coastguard 200” - Porcentagem de pacotes perdidos na comparação entre as técnicas <i>droptail</i> e ARED.	46
Figura 31 - Vídeo “Coastguard 1024” - Porcentagem de pacotes perdidos na comparação entre as técnicas <i>droptail</i> e ARED.	47

Figura 32 - Vídeo “News 200” - Porcentagem de quadros danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	48
Figura 33 - Vídeo “News 1024” - Porcentagem de quadros danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	48
Figura 34 - Vídeo “Coastguard 200” - Porcentagem de quadros danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	49
Figura 35 - Vídeo “Coastguard 1024” - Porcentagem de quadros danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	49
Figura 36 - Vídeo “News 200” - Porcentagem de quadros tipo I danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	50
Figura 37 - Vídeo “News 1024” - Porcentagem de quadros tipo I danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	50
Figura 38 - Vídeo “Coastguard 200” - Porcentagem de quadros tipo I danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	51
Figura 39 - Vídeo “Coastguard 1024” - Porcentagem de quadros tipo I danificados na comparação entre as técnicas <i>droptail</i> e ARED.....	51
Figura 40 - Atraso médio de rede e desvio padrão dos pacotes do vídeo “News 200”.....	52
Figura 41 - <i>Jitter</i> dos pacotes do vídeo “News 200”.....	53
Figura 42 - Atraso médio de rede e desvio padrão dos pacotes do vídeo “News 1024”.....	53
Figura 43 – <i>Jitter</i> dos pacotes do vídeo “News 1024”.....	54
Figura 44 - Atraso médio de rede e desvio padrão dos pacotes do vídeo “Coastguard 200”.....	54
Figura 45 – <i>Jitter</i> dos pacotes do vídeo “Coastguard 200”.....	55
Figura 46 - Atraso médio de rede e desvio padrão dos pacotes do vídeo “Coastguard 1024”.....	55
Figura 47 – <i>Jitter</i> dos pacotes do vídeo “Coastguard 1024”.....	56
Figura 48 - Goodput dos fluxos TCP sob a técnica <i>droptail</i>	57
Figura 49 - Goodput dos fluxos TCP sob a técnica ARED.....	57
Figura 50 - PSNR de 300 quadros da sequência “News 200” aplicando-se a técnica <i>droptail</i> com 10 fluxos TCP concorrentes.....	58
Figura 51 - PSNR de 300 quadros da sequência “News 200” aplicando-se a técnica ARED modo byte com 10 fluxos TCP concorrentes.....	59
Figura 52 - PSNR de 300 quadros da sequência “News 1024” aplicando-se a técnica <i>droptail</i> com 15 fluxos TCP concorrentes.....	59
Figura 53 - PSNR de 300 quadros da sequência “News 1024” aplicando-se a técnica ARED modo byte com 15 fluxos TCP concorrentes.....	60
Figura 54 - PSNR de 300 quadros da sequência “News 1024” aplicando-se a técnica ARED modo pacote com 15 fluxos TCP concorrentes.....	60
Figura 55 - PSNR de 300 quadros da sequência “Coastguard 1024” aplicando-se a técnica <i>droptail</i> com 10 fluxos TCP concorrentes.....	61
Figura 56 - PSNR de 300 quadros da sequência “Coastguard 1024” aplicando-se a técnica ARED modo pacote com 10 fluxos TCP concorrentes.....	61

RESUMO

Tendo em vista a crescente utilização de serviços de voz, áudio e vídeo transmitidos através da Internet - apesar da intrínseca inadequação do protocolo da Internet à transmissão de fluxos multimídia em tempo-real -, o presente trabalho avalia o impacto na qualidade de serviço oferecida pelas redes IP *best effort* que transportam fluxos de vídeo H.264/MPEG-4 AVC em tempo-real ao se aplicar a técnica de gerenciamento ativo de filas ARED em lugar da técnica *droptail* tradicional. As análises são feitas tanto do ponto de vista dos fluxos de vídeo isoladamente quanto do ponto de vista dos fluxos TCP que concorrem pelo mesmo enlace. Os resultados obtidos através de simulações indicam que a aplicação da técnica ARED reduz a quantidade de perdas de pacotes de vídeo porém danifica um número maior de quadros de vídeo do que a técnica *droptail*, enquanto que os indicadores de atraso médio e variação de atraso dos pacotes de vídeo mostram um pequeno ganho qualitativo.

Palavras-chave: H.264/MPEG-4 AVC sobre IP, qualidade de serviço, fluxos multimídia, transmissão de vídeo comprimido.

ABSTRACT

Considering the ever increasing utilization of voice, audio and video services transmitted through the Internet – despite the intrinsic inadequacy of the Internet protocol to the transmission of real-time multimedia streams -, this work evaluates the impact on the quality of service provided by best effort IP networks that transport H.264/MPEG-4 AVC real-time video streams when the ARED active queue management technique is used in place of the traditional droptail technique. The analysis are done both from the point of view of the video streams as well as from the point of view of the TCP streams that share the same bottleneck link. The results obtained through simulations indicate that the ARED technique reduces the loss of video packets but degrades a higher number of video frames if compared to the droptail technique, while delay and jitter show a small qualitative gain.

Keywords: H.264/MPEG-4 AVC over IP, quality of service, multimedia streams, transmission of compressed video.

1. INTRODUÇÃO

A transmissão de conteúdos multimídia em tempo-real através da Internet vem se tornando cada vez mais comum e consome mais recursos da rede mundial de computadores a cada dia que passa. Conseqüentemente, é cada vez maior a atenção dada à qualidade percebida pelos usuários, assim como é cada vez mais importante a preocupação com a influência que as aplicações multimídia em tempo-real exercem sobre os demais tipos de fluxos que trafegam pela Internet.

A crescente utilização dos serviços multimídia pode ser explicada por várias razões, tanto do ponto de vista dos provedores quanto dos usuários dos serviços. Os provedores tiram proveito da simplicidade e flexibilidade da comunicação através de redes comutadas a pacote que usam o protocolo da Internet (IP - *Internet Protocol*) para o transporte dos dados, e hoje disponibilizam uma variedade enorme de serviços através da rede mundial de computadores.

Do ponto de vista dos usuários, a primeira e mais importante razão diz respeito à popularização dos acessos em banda-larga, via redes de telefonia fixa ou redes sem fio, que permitem navegar na Internet a taxas relativamente altas e acessar conteúdos complexos num tempo razoável.

Além disso, os avanços nas tecnologias xDSL (*Digital Subscriber Line* – linha digital de assinante) aumentaram significativamente a capacidade de transmissão das redes de telefonia fixa sobre pares de cobre, obrigando as operadoras de telecomunicações a ampliarem a capacidade de transmissão de suas redes como um todo, do núcleo da rede à rede de acesso. A recente tecnologia GPON (*Gigabit-capable Passive Optical Network*), por sua vez, chega para aumentar em várias ordens de grandeza a capacidade da “última milha”, definitivamente eliminando desta a característica de “gargalo da rede” e permitindo aos usuários mais privilegiados acesso na faixa dos gigabits por segundo.

Também não se pode deixar de mencionar como razão para o crescimento da utilização de serviços multimídia o avanço nas tecnologias de compressão de voz, áudio e vídeo alcançado nos últimos anos. No caso particular da compressão de vídeo, a mais recente e significativa delas diz respeito ao padrão H.264/MPEG-4 AVC (AVC - *Advanced Video Coding*) [1].

Em termos comparativos, o padrão H.264/MPEG-4 AVC atinge o mesmo nível de qualidade percebida utilizando aproximadamente metade da banda (ou, dependendo da aplicação, metade do espaço de armazenamento) que o antigo padrão MPEG-2, usado nos DVDs e nos primeiros sistemas de TV digital, por exemplo. Visto por outro ângulo, uma mesma rede de acesso consegue fornecer serviços de vídeo com muito mais qualidade, se comparado ao padrão MPEG-2, graças à capacidade de compressão do H.264/MPEG-4 AVC.

Dentre os conteúdos audiovisuais complexos que compõem os chamados fluxos multimídia, os fluxos de voz em banda estreita (limitada a aproximadamente 3,5kHz) e de vídeo codificado no padrão MPEG-4 já são bastante comuns na Internet, respectivamente na forma de serviços VoIP (*Voice over Internet Protocol* - voz sobre o protocolo da Internet) e serviços de *Internet TV* (televisão através da Internet).

Cabe aqui uma importante distinção entre os serviços de *Internet TV* e IPTV (*Internet Protocol Television* – televisão sobre o protocolo da Internet): ambos são oferecidos sobre redes IP, porém o IPTV prevê a utilização de *set-top-boxes* IP fornecidos pelo provedor do serviço – que normalmente também controla a rede, o conteúdo distribuído e a QoS (*Quality of Service* – qualidade do serviço) [2]. A Figura 1 mostra um sistema de IPTV típico.

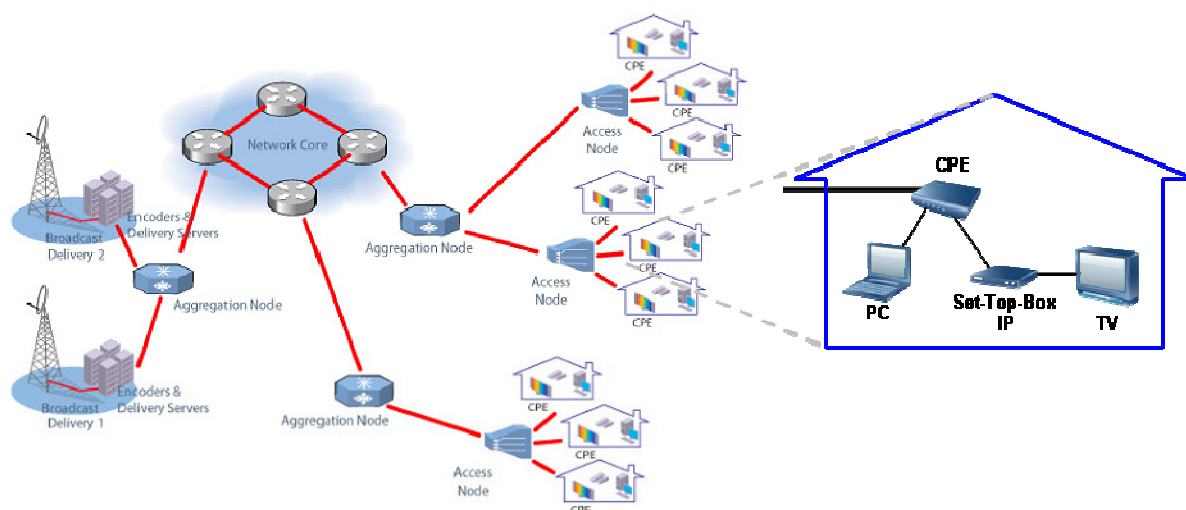


Figura 1 - Topologia típica de um sistema de IPTV [3].

Já o serviço de *Internet TV* engloba conteúdo gerado de qualquer ponto da Internet e que pode ser assistido ou baixado pelo usuário final utilizando geralmente um PC (*Personal Computer* – computador pessoal) [2], conforme mostra a Figura 2. Este trabalho enfoca principalmente o serviço de transmissão de vídeo em tempo-real através de uma rede IP *best effort*, caracterizando o conteúdo visual do serviço de *Internet TV*.

No entanto, dentro do atual modelo de serviço *best effort* (melhor esforço) que a Internet oferece, não se podem dar garantias quanto a perdas e atrasos dos pacotes transmitidos, nem quanto à largura de banda disponível, porém pode-se procurar minimizar o impacto altamente nocivo que as perdas e os atrasos provocam na QoS de aplicações multimídia em tempo-real. Como perdas normalmente são provocadas pelo descarte de pacotes em nós de rede que enfrentam congestionamento, onde o método de tratamento das filas é, na maioria das vezes, o

chamado *droptail*, a implementação de técnicas de gerenciamento ativo de filas pode proporcionar ganhos qualitativos importantes, dependendo das características dos fluxos transportados pela rede em questão, conforme demonstrado por Vitalio *et al.* no que diz respeito a fluxos VoIP [4].

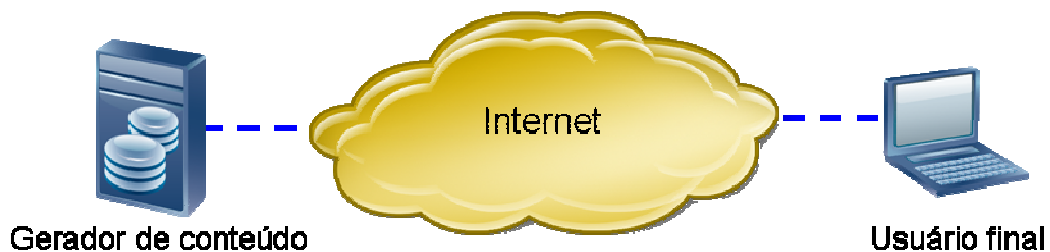


Figura 2 - Topologia de um sistema de Internet TV.

Sendo assim, o presente trabalho avalia a influência em termos de quantidade de pacotes perdidos, quantidade de quadros danificados, atraso médio de rede e variação do atraso (*jitter*) que os fluxos de vídeo H.264/MPEG-4 AVC sofrem a partir da aplicação da técnica de gerenciamento ativo de fila (AQM – *Active Queue Management*) do tipo detecção adiantada aleatória na sua versão adaptativa (ARED - *Adaptive Random Early Detection*). Os resultados são então comparados aos obtidos com a aplicação da técnica *droptail* tradicional. A topologia proposta foi simulada no software Network Simulator versão 2 (NS2) [5].

2. CONCEITOS FUNDAMENTAIS E REVISÃO BIBLIOGRÁFICA

2.1 A Internet como Meio de Transmissão

A Internet pode ser definida como um conjunto de diversas redes interconectadas que utilizam o protocolo IP na camada de rede para a transmissão de pacotes de informação (ou datagramas). É um sistema incomum que não foi planejado por ninguém e não é controlado por ninguém, e qualquer pessoa ou organização pode acessá-la através de uma gama enorme de aplicações.

Se por um lado a camada de rede da Internet pode ser considerada “o pino que mantém a arquitetura da Internet unida” [6], por outro trata-se apenas de uma das sete camadas que compõem o modelo OSI (*Open System Interconnection*), conforme mostra a Figura 3.

Aplicação
Apresentação
Sessão
Transporte
Rede
Enlace de Dados
Física

Figura 3 - Hierarquia de protocolos do modelo OSI.

2.2 Exemplo de operação do IP

A Figura 4 mostra um exemplo típico da operação do IP, onde duas LANs (*Local Area Networks* – Redes locais) são interconectadas por uma rede WAN (*Wide Area Network*) do tipo *frame-relay*. O *host* PC 1, localizado na LAN 1, troca informações com o *host* PC 2, localizado na LAN 2, através da rede *frame-relay*. A figura mostra a arquitetura de protocolos em cada etapa. Tanto os PCs 1 e 2 quanto os roteadores precisam compartilhar um protocolo em comum, o IP. Além disso, os PCs 1 e 2 precisam compartilhar os mesmos protocolos nas camadas acima do IP. Já os roteadores intermediários precisam implementar os protocolos da camada física até o IP [7].

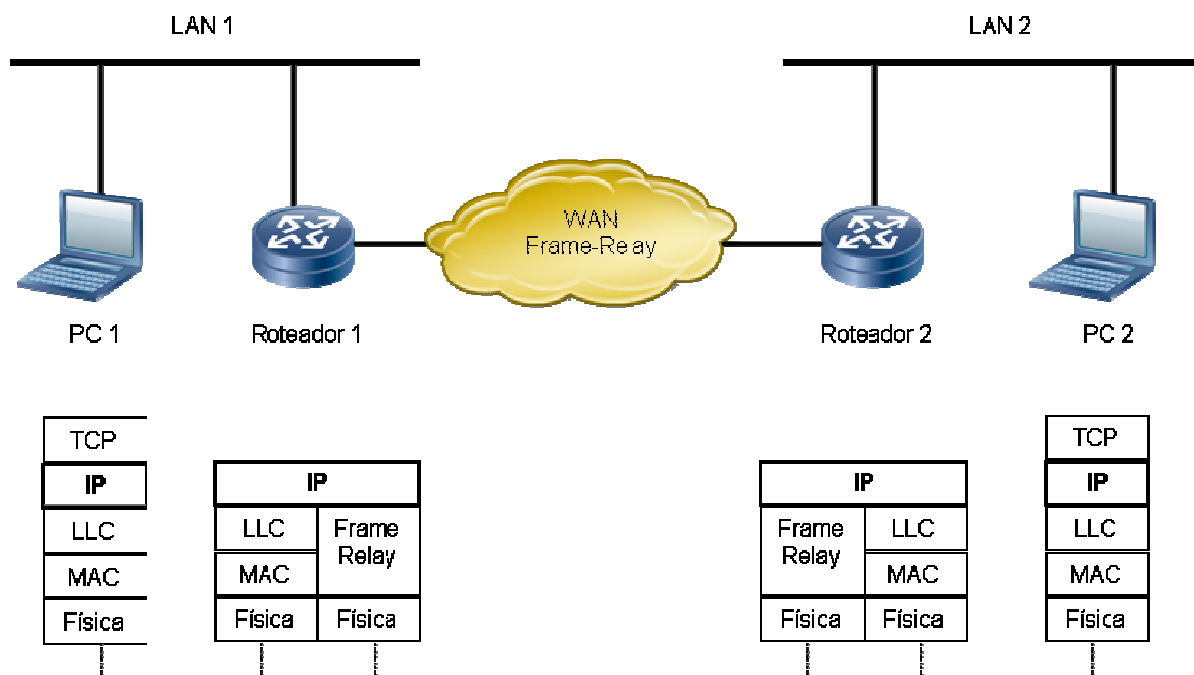


Figura 4 - Exemplo de operação do protocolo IP.

O serviço oferecido pelo IP não é confiável, ou seja, não há garantia de que todos os dados serão entregues ou de que serão entregues na ordem correta. É responsabilidade da camada imediatamente superior (TCP, neste exemplo) recuperar qualquer erro que ocorra.

Com o IP, cada datagrama é passado de roteador para roteador, numa tentativa de se chegar ao destinatário final. Como a entrega não é garantida, não existem exigências de confiabilidade sobre qualquer uma das redes que compõem a “internet” (o termo “internet” aqui tem o sentido genérico de “entre-redes”). Dessa forma, o IP funcionará com qualquer combinação de tipos de rede. Como a sequência de entrega também não é garantida, datagramas podem seguir diferentes caminhos através da “internet”, o que permite ao protocolo reagir tanto a congestionamentos quanto a falhas nas redes através da mudança de rotas.

2.3 Característica *Best Effort* da Internet

Best effort descreve uma modalidade de serviço em que a rede não oferece nenhuma garantia de que os dados serão entregues, ou de que seja dada a qualquer usuário algum nível garantido de qualidade de serviço ou prioridade. Numa rede *best effort* todos os usuários recebem tratamento idêntico, o que significa taxa de bits e tempo de transmissão variáveis, que dependem da carga de tráfego naquele determinado momento. Ao remover mecanismos de

recuperação de dados perdidos ou corrompidos e pré-alocação de recursos, a rede pode operar de forma mais eficiente e possuir nós mais simples e baratos.

Roteadores IP convencionais operam dentro do modelo *best effort*. A simplicidade desse modelo é o fator chave do maior sucesso do IP em relação a protocolos mais complexos como X.25 e ATM (*Asynchronous Transfer Mode*).

Apesar de ser essencialmente uma rede de comunicação *best effort*, pode-se oferecer garantias de entrega de dados na Internet através de protocolos de camadas superiores, implementados nos elementos finais, ao invés dos elementos nos nós da rede. Esse é o caso do protocolo TCP, que garante a entrega dos dados no receptor através de mecanismos de retransmissão. A existência de tais mecanismos, no entanto, dificulta a utilização do TCP para a transmissão de dados em tempo-real. Já o UDP não oferece tal garantia, caracterizando um transporte de dados do tipo *best effort* mais adequado à transmissão de dados em tempo-real.

2.4 Paradigma de Tráfego TCP

O paradigma de tráfego TCP determina que um transmissor reduza sua taxa de transmissão de bits pela metade (resultado do ajuste do tamanho do buffer TCP) tão logo a perda de pacotes ultrapasse certo limiar. Uma vez que a perda de pacotes caia abaixo desse limiar (mais um determinado valor de histerese) o transmissor pode aumentar lentamente sua taxa de bits, até que a taxa de perdas seja novamente alta demais e todo o processo recomeça. Esse mecanismo simples, porém efetivo, evita a sobrecarga dos roteadores e assegura uma razoável “justiça” entre todos os transmissores. Deve-se enfatizar que, devido a essa arquitetura, a perda de pacotes e as taxas de transmissão ditadas pela própria rede são características básicas de todas as redes *best effort*, e não o resultado de algum tipo de falha na rede. Esse mecanismo foi introduzido como um método de controle de congestionamento e funciona bem se as perdas são causadas por congestionamento [8].

2.5 Transmissão de Dados Através da Combinação UDP/IP

Quando a característica *best effort* do IP e os mecanismos de confiabilidade de transmissão do TCP são combinados, a utilização do TCP/IP para a transmissão de fluxos em tempo-real pode não ser a mais adequada pois tais mecanismos são altamente dependentes dos atrasos na rede. Como uma parcela significativa das aplicações multimídia presentes na Internet, tais como VoIP, fluxos de áudio e vídeo em tempo-real e videoconferências, possuem características conversacionais e restrições muito fortes em termos de tempo de transmissão e exigência de largura de banda, tornou-se necessária a utilização de uma outra combinação de protocolos.

Surge então a combinação entre o UDP e o IP, onde a confiabilidade e o “bom comportamento” do TCP dão lugar à simplicidade do UDP, que apenas envia mensagens com um mínimo de mecanismos envolvidos, sem garantia de entrega nem proteção contra entregas duplicadas. Basicamente, o que o UDP faz é, do lado do transmissor, repassar ao IP segmentos de dados de uma determinada aplicação para que este faça o roteamento até o destino e, do lado do receptor, repassar os segmentos do IP para a aplicação [9]. Com o UDP, se esquemas adicionais de controle de erro forem necessários, terão de ser implementados numa camada superior [8].

O UDP, no entanto, não foi originalmente criado para o transporte de voz, áudio e vídeo em tempo-real na Internet, ele é apenas a melhor escolha em lugar do TCP. Para que o UDP seja utilizado no tráfego de conteúdos multimídia, é desejável o emprego de outros protocolos na camada de aplicação. Com essa finalidade foi desenvolvido o RTP que oferece serviços de entrega fim-a-fim para dados com características de tempo-real. Esses serviços incluem a identificação do tipo de dado transportado, número de seqüência, marcadores de tempo e monitoramento de entrega. Normalmente as aplicações rodam o RTP acima do UDP para fazer uso dos seus serviços de multiplexação, verificação de erros nos dados e re-ordenamento de pacotes [9].

A especificação do RTP prevê um protocolo de controle simples que o acompanha, chamado RTCP (*Real-Time Control Protocol*). Esse protocolo pode ser empregado para permitir a troca periódica de informações de controle entre os participantes de uma sessão [9] e informar ao codificador a respeito das características da rede e do caminho de transmissão sob o ponto de vista do decodificador, de forma que um codificador otimizado para tal uso possa adaptar suas características de resistência a erros e taxa de transmissão de bits em tempo-real [8].

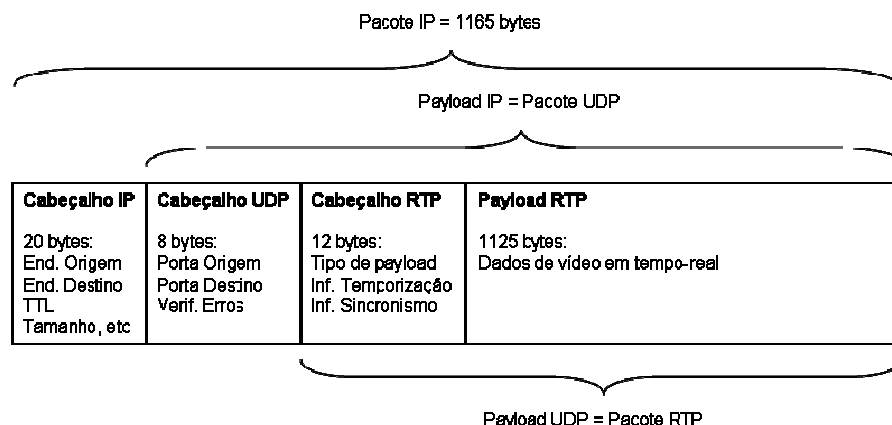


Figura 5 - Exemplo de encapsulamento RTP sobre UDP sobre IP.

A Figura 5 mostra como fica o encapsulamento RTP sobre UDP sobre IP, tomando como exemplo um segmento de informação de vídeo de 1125 bytes. Já a Figura 6 mostra esquematicamente a interação entre pacotes de vídeo enviados conforme o encapsulamento acima e o protocolo RTCP.

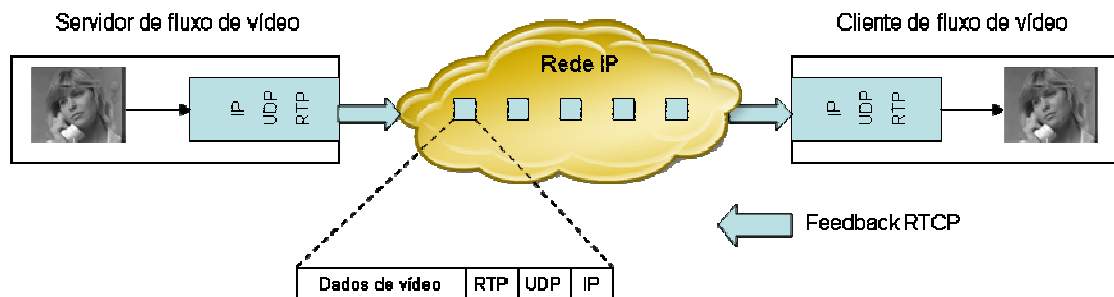


Figura 6 - Exemplo de arquitetura de transmissão de fluxo de vídeo em tempo-real.

2.6 Característica Inelástica dos Dados em Tempo-Real

O tráfego numa “internet” (novamente, o termo “internet” aqui tem o sentido genérico de “entre-redes”) pode ser dividido em duas categorias principais: elástico e inelástico [7].

Tráfego elástico é aquele que consegue se ajustar às mudanças em termos de atraso e *throughput* ao atravessar uma “internet” e ainda assim atender às necessidades de suas aplicações. Esse é o tipo de tráfego tradicional suportado em redes e “internets” TCP/IP, e é o tipo de tráfego para o qual tais redes foram projetadas. Aplicações que geram tal tipo de tráfego geralmente usam TCP ou UDP como protocolo de transporte. No caso do UDP, a aplicação usará tanta capacidade quanto possível, até o limite da taxa de geração de dados da aplicação. No caso do TCP, a aplicação usará tanta capacidade quanto possível, até o limite da taxa de recepção dos receptores finais. Além disso, o TCP ajusta o tráfego em conexões individuais de acordo com o congestionamento, através da redução da taxa em que os dados são inseridos na rede. Aplicações que podem ser classificadas como elásticas incluem aplicações comuns que operam sobre TCP ou UDP, cujos principais exemplos são: transferência de arquivos (FTP), correio eletrônico (SMTP), *login* remoto (Telnet), gerenciamento de rede (SNMP) e acesso Web (HTTP). Entretanto, existem diferenças quanto às necessidades dessas aplicações. Por exemplo:

- Aplicações de e-mail geralmente são bastante insensíveis a variações no atraso.

- Quando a transferência de arquivos é feita interativamente, como geralmente acontece, o usuário espera que a demora seja proporcional ao tamanho do arquivo, portanto torna-se sensível a variações no *throughput*.
- Quanto ao gerenciamento de redes, o atraso normalmente não é uma preocupação séria. No entanto, se falhas na rede são causa de congestionamento, então a necessidade de mensagens SNMP serem transmitidas através da rede com um mínimo de atraso aumenta com o aumento do congestionamento.
- Aplicações interativas, tais como *login* remoto e acessos Web, são bastante sensíveis a atrasos.

Tráfego inelástico é aquele que não se adapta facilmente, ou absolutamente não se adapta, a variações de atraso e *throughput* numa “internet”. O principal exemplo é o tráfego em tempo-real. As necessidades do tráfego do tipo inelástico podem incluir:

- *Throughput*: pode ser necessário um *throughput* mínimo. Diferente da maioria dos tráfegos elásticos, que conseguem enviar dados mesmo com um serviço degradado, muitas aplicações inelásticas requerem um dado *throughput* mínimo.
- Atraso: exemplos típicos de aplicações sensíveis ao atraso são as de mercados de ações. Alguém que recebe o serviço com atraso vai atuar com atraso, e com grande desvantagem.
- *Jitter*: a magnitude da variação do atraso é um fator crítico para aplicações em tempo-real. Quanto maior a variação de atraso permissível, maior o atraso na entrega dos dados e maior o tamanho do buffer de atraso necessário nos receptores. Aplicações interativas em tempo-real podem exigir um limite superior de *jitter*.
- Perdas de pacotes: a quantidade de pacotes perdidos que aplicações em tempo-real podem tolerar varia, ou mesmo pode ser nula.

Tais necessidades são difíceis de atender num ambiente com atrasos de enfileiramento variáveis e perdas provocadas por congestionamento. Surge daí a introdução de duas novas características na arquitetura da Internet. A primeira diz respeito à necessidade de prover tratamento preferencial a aplicações com maiores exigências. Essa abordagem implica o uso de protocolos de reserva de recursos.

A segunda característica diz respeito à necessidade de suportar tráfego inelástico e elástico simultaneamente numa arquitetura de “internet”. Aplicações inelásticas geralmente não reduzem a taxa em face de congestionamentos, ao contrário de aplicações baseadas no TCP. Portanto, em períodos de congestionamento, o tráfego inelástico continuará gerando uma carga alta, e ao tráfego elástico caberá uma parcela cada vez menor da banda disponível [7].

2.7 Técnicas de Controle de Congestionamento

Dentro do paradigma de tráfego TCP, a perda de pacotes que provoca a redução na taxa de envio por parte do transmissor normalmente é causada pelo descarte de pacotes que os roteadores congestionados fazem ao terem seus *buffers* de enfileiramento completamente cheios. Essa técnica de descarte para controle de congestionamento é chamada *drop-tail*. No entanto, o tráfego UDP não responde da mesma maneira que o TCP à perda de pacotes. Na verdade, não há resposta alguma (tráfego irresponsivo), o que provoca uma divisão “injusta” entre tráfego TCP e tráfego UDP na banda de transmissão disponível [4].

Técnicas de gerenciamento ativo de filas, mais eficientes e justas no controle do congestionamento de redes IP, vêm sendo propostas por diversos autores e os resultados apresentados parecem promissores, especialmente quanto aos parâmetros de atraso e *jitter*, tão importantes para aplicações multimídia em tempo-real.

Vitalio *et al.* [4] apresentam um extenso estudo sobre o impacto do gerenciamento ativo de filas na qualidade de serviço das aplicações de VoIP. Analisa-se um dos algoritmos AQM mais representativos através de simulação computacional, avaliando seu efeito na qualidade de voz percebida. Diferentes cenários de rede são investigados alterando a carga de tráfego e os parâmetros de controle da rede. A degradação da transmissão é vinculada à percepção dos usuários por meio de modelos teóricos bem conhecidos, expressando a satisfação destes na escala MOS (Mean Opinion Score). Os principais resultados obtidos mostram que o uso de mecanismos para o gerenciamento ativo de filas, tais como ARED, melhora significativamente a qualidade de voz percebida. Este estudo é ampliado em [11] incorporando outros mecanismos de controle de congestionamento tais como PI, REM e AVQ.

2.8 RED e ARED

Tratar o congestionamento logo que o mesmo é detectado, ao invés de deixá-lo crescer ao ponto de interromper os serviços da rede, é sem dúvida a melhor estratégia para abordar o problema. Tal observação leva à idéia de descartar pacotes antes que todo o espaço de armazenamento seja preenchido. Um algoritmo popular que implementa esse conceito chama-se RED (*Random Early Detection* – Detecção Adiantada Aleatória) [22], tendo sido o primeiro algoritmo de AQM proposto.

Em alguns protocolos de transporte, como o TCP, a resposta dada pela fonte de dados é reduzir a taxa de transmissão. A razão por trás dessa lógica é que o TCP foi projetado para operar em redes cabeadas, que são muito confiáveis e onde a perda de pacotes normalmente é

provocada pelo transbordamento dos *buffers*, ao invés de erros de transmissão. Esse fato pode ser explorado e usado na redução do congestionamento.

A idéia por trás do descarte de pacotes antes que a situação torne-se crítica (daí o nome “detecção adiantada”) é ter tempo de ação antes que seja tarde demais. Para determinar quando começar a descartar, os roteadores calculam a média corrente do tamanho das suas filas. Quando o tamanho médio da fila ultrapassa um determinado limite, considera-se a linha congestionada e tomam-se ações para diminuir o congestionamento. Como os roteadores normalmente não conseguem distinguir qual é a fonte de dados que está provocando o congestionamento, descarta-se aleatoriamente.

Uma das formas de comunicar à fonte de dados sobre o congestionamento é enviar um pacote “choke”. O problema desse mecanismo é que tais pacotes só fazem aumentar a carga numa rede já congestionada. Uma estratégia diferente é simplesmente descartar o pacote selecionado e não reportar a ação. A fonte de dados perceberá a falta da confirmação da recepção e tomará a ação de reduzir a taxa, ao invés de insistir na transmissão, pois sabe que perdas de pacotes são normalmente provocadas por congestionamentos. Essa forma implícita de realimentação somente funciona quando a fonte responde à perda de pacotes através da redução da taxa de transmissão. Em redes sem fio, onde a maioria das perdas é provocada por ruídos no meio de transmissão, esse mecanismo não deve ser usado [6].

A técnica RED exige a configuração de quatro parâmetros: limiares mínimo e máximo de tamanho de fila (\min_{th} e \max_{th}), probabilidade máxima de marcação (\max_p) e constante de fila média (w_q). Variações nos valores desses parâmetros afetam dramaticamente o desempenho, e configura-los corretamente é uma tarefa bastante difícil. Além do mais, é praticamente impossível encontrar um conjunto de valores que funcionem em todos os cenários existentes na Internet. Outro elemento que impacta o desempenho da técnica RED é o modo de medição do tamanho da fila, se em bytes ou em pacotes. Pacotes grandes têm maior probabilidade de serem marcados quando operando em modo byte. Isso significa que aplicações que geram pacotes pequenos têm tratamento preferencial no modo byte [4].

A técnica RED aumenta a sinalização enviada à fonte do congestionamento proporcionalmente ao tamanho médio da fila. A probabilidade de marcação de pacotes $p(a_q)$ é computada como

$$p(a_q) = \begin{cases} 0 & \text{se } a_q < \min_{th} \\ 1 & \text{se } a_q \geq \max_{th} \\ \max_p \cdot \left(\frac{a_q - \min_{th}}{\max_{th} - \min_{th}} \right) & \text{demais casos} \end{cases} \quad (1)$$

onde a_q é a média ponderada exponencial móvel (EWMA – *Exponential Weighted Moving Average*) do tamanho da fila, e é atualizada a cada novo pacote que chega amostrando-se o tamanho da fila naquele momento (q) como

$$a_q \leftarrow (1 - w_q) \cdot a_q + w_q \cdot q \quad (2)$$

ARED é uma versão aprimorada da técnica RED, que possui a habilidade de ajustar dinamicamente os parâmetros às condições correntes de tráfego [23]. Em resumo, a abordagem dos autores é adaptar \max_p de forma a manter o tamanho médio da fila entre \min_{th} e \max_{th} , com as seguintes particularidades:

- \max_p é adaptado não só para manter o tamanho médio da fila entre \min_{th} e \max_{th} , mas de forma a manter o tamanho médio da fila dentro de uma faixa-alvo centralizada entre \min_{th} e \max_{th} ;
- \max_p é adaptado lentamente, sobre escalas de tempo maiores que um RTT (*Round-Trip Time* – tempo de ida e volta) típico, e a passos pequenos;
- \max_p é restrito à faixa de valores [0,01 e 0,5] (ou equivalentemente [1%, 50%]);
- Ao invés de aumentar e diminuir \max_p multiplicativamente, usa-se uma política de aumentar aditivamente e diminuir multiplicativamente \max_p .

Floyd *et al.* [23] não afirmam que o algoritmo ARED implementado por eles seja ótimo ou mesmo próximo de ótimo, mas que o mesmo parece funcionar bem numa grande variedade de cenários, e que poderia ser implementado com segurança na Internet. De acordo com seus projetistas, o único parâmetro que precisa ser estabelecido nessa técnica é o tamanho alvo da fila ou, equivalentemente, o atraso alvo de enfileiramento. Um estudo recente demonstrou que ARED possui efeito positivo no tráfego Web quando operando em modo byte [4].

No presente trabalho foi dada especial atenção à aplicação da técnica ARED porque esta mostrou um expressivo ganho qualitativo na transmissão de fluxos VoIP, os quais, assim como fluxos de vídeo em tempo real, possuem sérias restrições em termos de perdas de pacotes de informação e temporização [4] [11].

2.9 Transmissão de Vídeo Comprimido em Redes IP

Tecnologias de compressão de vídeo são usadas em diversas aplicações, pois reduzem significativamente as exigências de espaço para armazenamento e largura de banda para transmissão. No entanto, trazem consigo o problema da degradação da qualidade percebida.

No caso da transmissão de vídeo comprimido através de redes IP, a degradação da qualidade pode ser classificada em duas categorias. A primeira categoria de degradação

perceptual é causada pelo processo de compressão de vídeo propriamente dito (efeito “mosaico”, bordas borradas, bordas quadriculadas, etc. mostrados na Figura 7) [12]. A segunda categoria de degradação perceptual é causada pela inevitável perda de pacotes nas atuais redes IP *best effort*, cujo efeito é mostrado na Figura 8 [12].

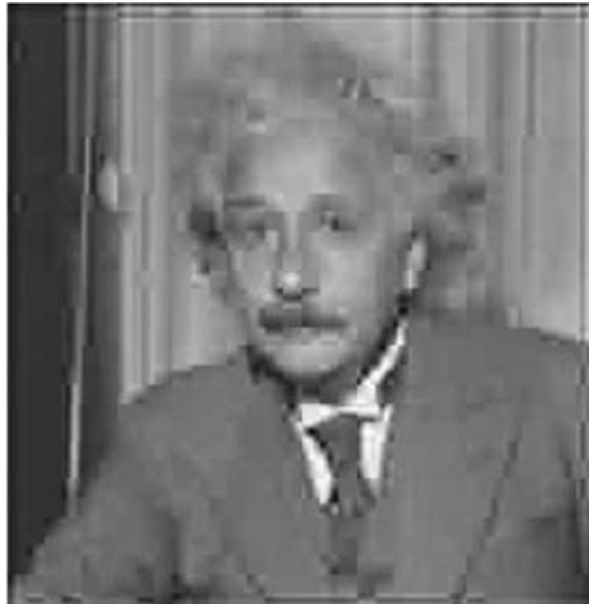


Figura 7 - Exemplo do efeito de bordas quadriculadas causado pela compressão.



Figura 8 - Exemplo do efeito causado por perdas de pacotes na transmissão.

O tráfego de vídeo é uma fração importante e de crescimento rápido dentre os diversos tipos de tráfego suportados por redes IP. Impulsionado pelo sucesso de aplicações multimídia como a transmissão de fluxos de vídeo e vídeo-conferência, espera-se que o tráfego de vídeo repita o sucesso que outros tipos de tráfego multimídia em tempo-real sobre IP atingiram, como é o caso da transmissão de voz (VoIP) e áudio (MP3 – MPEG-1 Layer III) [13].

No entanto, o tráfego de vídeo enfrenta os mesmos problemas de QoS que todas as outras modalidades de tráfego multimídia em tempo-real enfrentam nas redes IP: não há garantias de controle de atraso, *jitter* e perdas de pacotes, sendo este último um sério problema para fluxos de vídeo com altas taxas de compressão [13] [14].

Adicionado a isso, o crescente peso do tráfego multimídia baseado no protocolo UDP vem levantando preocupações por seus efeitos sobre as conexões TCP concorrentes, preocupações essas que são particularmente importantes no caso do tráfego de vídeo, consumidor de grande largura de banda [13].

Várias soluções têm sido propostas para o tratamento desses problemas. No nível da arquitetura da rede, modelos que introduzem QoS nas redes IP vêm sendo discutidos, notadamente o Modelo de Serviços Diferenciados (*DiffServ*). No nível de aplicação diversas técnicas para se melhorar a QoS têm sido propostas, incluindo FEC (*Forward Error Correction*), particionamento de dados, *error concealment*, retransmissão seletiva de pacotes e vários tipos de codificação adaptativa [13]. Este trabalho avalia o impacto em termos de pacotes de vídeo perdidos, quadros de vídeo danificados, atrasos e *jitter*, a partir da implementação da técnica ARED de AQM nos roteadores que formam o gargalo de uma rede simulada.

2.10 Taxa de Bit Constante versus Taxa de Bit Variável

Existem duas maneiras principais para se codificar sinais de vídeo [19]:

- a) com escalas de quantização fixas, resultando em qualidade praticamente constante às custas de uma taxa de bits variável;
- b) com controle de taxa, que adapta as escalas de quantização de forma a manter a taxa de transmissão praticamente constante às custas de uma qualidade variável.

Este trabalho enfoca a codificação de vídeo usando escalas de quantização fixas a fim de se examinar as características fundamentais de tráfego do padrão H.264/MPEG-4 AVC.

2.11 Principais Aplicações da Transmissão de Vídeo na Internet

A natureza das transmissões de vídeo na Internet determina as restrições e o ambiente do protocolo com o qual a fonte de vídeo tem de conviver. Ao usar IP na camada de transporte, três tipos fundamentais de aplicações podem ser identificadas, e são resumidamente comentadas a seguir [8]:

1) Aplicações conversacionais (videotelefonia e videoconferência): tais aplicações se caracterizam por restrições de atraso muito rígidas, com atraso fim-a-fim de muito menos de um segundo.

2) *Download* de fluxos de vídeo pré-gravados: neste caso os fluxos de bits são transmitidos como um todo, usando protocolos confiáveis tais como FTP ou HTTP, e não há restrições em termos de atrasos e resiliência a erros.

3) Transmissão de fluxos baseados em IP: essa é uma tecnologia que, no que diz respeito às características de atraso, está no meio do caminho entre aplicações conversacionais e *download* de fluxos. Não há uma definição amplamente aceita do termo “transmissão de fluxos”. Muitos o associam a um serviço de transmissão que permite começar a passar o vídeo antes que a transmissão da totalidade do fluxo de bits tenha sido concluída, com um atraso inicial de apenas alguns segundos, e em modo quase tempo-real. O fluxo de vídeo pode ser tanto pré-gravado e transmitido sob demanda, quanto uma sessão ao vivo pode ser comprimida em tempo-real e enviada a certo número de usuários através de um ou mais canais *multicast*. Devido a restrições de atraso “relaxadas” quando comparadas a serviços conversacionais, algumas ferramentas de codificação de vídeo de alto atraso podem ser usadas. No entanto, sob condições normais, os serviços de transmissão de fluxos usam protocolos de transmissão não-confiáveis (normalmente RTP/UDP/IP), tornando o controle de erros na fonte e/ou no canal uma necessidade.

2.12 Conceitos Básicos da Codificação de Vídeo

Uma imagem digital, ou um quadro de vídeo digital, tipicamente consiste de três matrizes retangulares de amostras, uma matriz para cada componente de uma representação “tristimulus” das cores na área que representa a imagem. A codificação de vídeo geralmente representa as cores através de três componentes, chamadas Y, Cb e Cr. A componente Y é chamada luma e representa o brilho. As duas componentes chroma Cb e Cr representam o quanto a cor se desvia do cinza em direção ao azul e vermelho, respectivamente. Como o sistema visual humano é mais sensível a luma do que a chroma, geralmente utiliza-se uma estrutura de amostragem em que as matrizes das componentes chroma têm apenas um quarto da quantidade de amostras da matriz da componente luma (metade da quantidade de amostras

tanto na horizontal quanto na vertical). Esse tipo de amostragem é chamado de 4:2:0. A amplitude de cada componente é tipicamente representada com precisão de 8 bits por amostra. Os parâmetros acima são referências para vídeo em qualidade padrão “consumidor”. Aplicações mais exigentes podem requerer diferentes combinações de quantidade de amostras e quantidade de bits por amostra [15].

Há dois formatos básicos de vídeo: progressivo e entrelaçado. Pode-se dizer que uma matriz de amostras de vídeo contém dois campos entrelaçados, um superior e outro inferior, conforme mostrado na Figura 9. O campo superior contém as linhas pares (0, 2, ..., H-2), onde 0 é a primeira linha e H é o número total de linhas. O campo inferior contém as linhas ímpares (começando com a segunda linha). Quando o entrelaçamento é usado, ao invés de capturar toda a imagem de uma só vez, apenas um dos campos é capturado. Portanto, são necessários dois períodos de amostragem para se capturar a imagem por completo. Se os dois campos de uma imagem são capturados em instantes diferentes, a imagem é chamada de entrelaçada. Caso contrário, a imagem é chamada de progressiva [15].

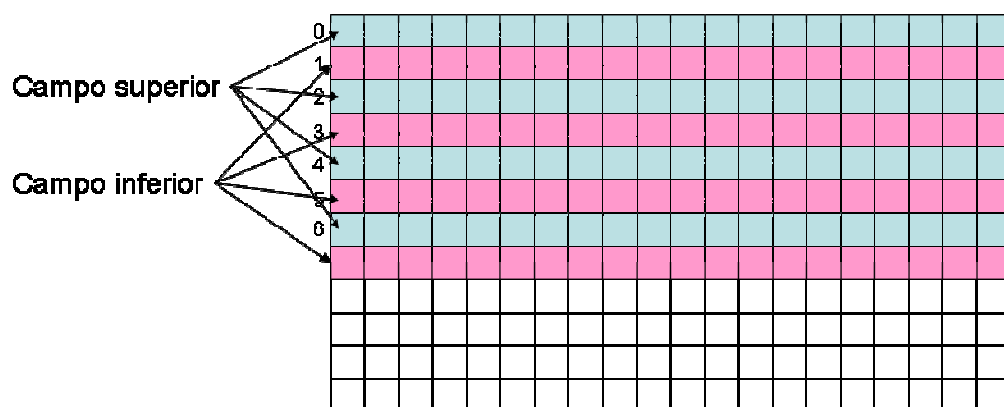


Figura 9 - Campos envolvidos nos dois formatos básicos de vídeo.

Técnicas para compressão de vídeo:

a) Predição: processo através do qual cria-se um conjunto de valores de predição (normalmente em parte baseados em indicações dadas pelo codificador sobre como formar a predição baseando-se na análise das amostras e tipos de predição que podem ser escolhidos) que são usados para se predizerem os valores das amostras, de forma que os valores a serem representados tornam-se apenas as diferenças para os valores preditos, ou também chamados de valores residuais [15].

b) Transformada: processo que está intimamente relacionado com a predição, e que consiste em formar um novo conjunto de amostras a partir de uma combinação de amostras de

entrada, geralmente usando combinação linear. De uma forma simplificada, pode-se dizer que a transformada elimina a necessidade de se representar repetidamente valores similares, podendo capturar a essência do sinal de entrada através da análise no domínio da frequência. Um benefício típico da transformada é a redução na correlação estatística das amostras de entrada, de forma que os aspectos mais relevantes do conjunto de amostras de entrada estão tipicamente concentrados num menor número de variáveis. A transformada discreta do cosseno (DCT - *Discrete Cosine Transform*) é um exemplo de transformada que tem um desempenho ótimo quando aplicada a fontes auto-regressivas altamente correlacionadas [15].

c) Quantização: processo através do qual a precisão utilizada para se representar o valor da amostra é reduzido a fim de se reduzir a quantidade de dados necessários para se representar a codificação. Tal processo é análogo ao conceito de se arredondar os dígitos menos significativos. A quantização é um processo não-reversível, pois necessariamente envolve alguma forma de mapeamento “muitos para poucos”, e que inerentemente envolve alguma perda de fidelidade [15].

d) Codificação por entropia: processo através do qual símbolos de valores discretos são representados de forma a tirar vantagem da probabilidade relativa de ocorrência dos possíveis valores de cada símbolo. Um tipo conhecido de codificação por entropia é a codificação por tamanho variável (VLC - *Variable-Length Coding*), que estabelece uma tabela de codificação em forma de árvore que usa palavras binárias curtas para representar símbolos que ocorrem com maior probabilidade, e palavras binárias mais longas para símbolos com menor probabilidade de ocorrência [15].

2.13 O Padrão H.264 / MPEG-4 *Advanced Video Coding* (H.264/AVC)

H.264/AVC é a denominação da mais recente recomendação de compressão de vídeo desenvolvida em conjunto pelos grupos internacionais de estudos VCEG da ITU-T e MPEG da ISO/IEC.

Atingindo taxas de compressão tipicamente 50% menores que o padrão MPEG-2 para um mesmo nível de qualidade percebida, o padrão H.264/AVC reúne o que há de mais moderno nas técnicas de codificação e decodificação de vídeo. A dramática melhora na taxa de codificação fez do H.264/AVC o padrão escolhido na maioria dos novos produtos e serviços de transmissão e armazenamento de vídeo, desde a transmissão de serviços de vídeo sobre linhas DSL, passando pela radiodifusão de vídeo para dispositivos móveis sem-fio e pelos mais modernos tocadores de vídeo usados na Internet, chegando até os padrões de discos ópticos BluRay e HD-DVD [16] [17].

O padrão H.264/AVC oferece eficiência de codificação significativa, especificações de sintaxe simples e fácil integração da codificação de vídeo em todas as arquiteturas de protocolos atuais. Possui os mesmos elementos funcionais básicos de padrões anteriores (MPEG-1, MPEG-2 parte 2, H.261, H.263), isto é, uso da transformada discreta de cossenos para redução da correlação espacial, quantização para controle da taxa de bits, predição com compensação de movimento para redução da correlação temporal e codificação de entropia para redução da correlação estatística.

No entanto, mudanças importantes foram introduzidas no padrão H.264/AVC a fim de melhorar o desempenho do codificador tais como: predição *intra-picture*, uma nova transformada 4x4 com números inteiros, figuras com múltiplas referências, blocos de tamanhos variáveis e compensação de movimento com precisão de $\frac{1}{4}$ de pixel, filtro de *deblocking* e melhoras na codificação de entropia [1].

Além das melhorias acima, o padrão H.264/MPEG-4 AVC define vários perfis de acordo com características tanto da aplicação quanto da plataforma, variando de plataformas com baixo poder de processamento em ambientes com alta taxa de perdas (perfil *baseline*), até aplicações profissionais que apenas suportam codificação Intra. A Figura 10 e a Figura 11 mostram, respectivamente, o diagrama de blocos do codificador e do decodificador H.264/AVC [1].

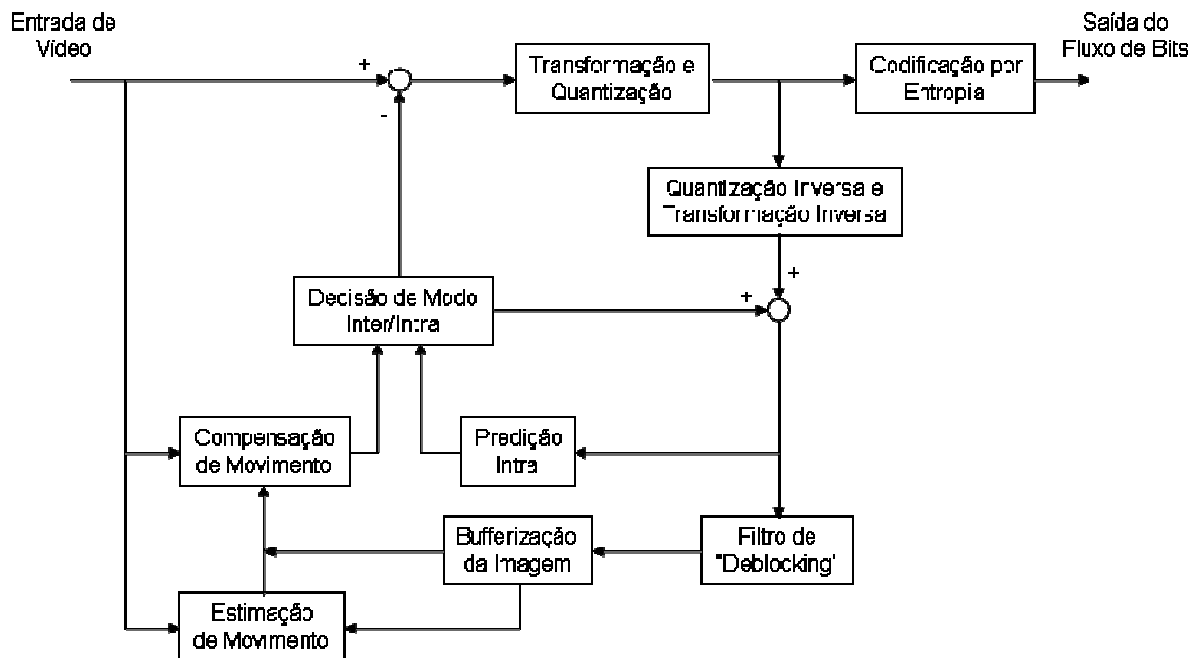


Figura 10 - Diagrama de blocos do codificador H.264/AVC.

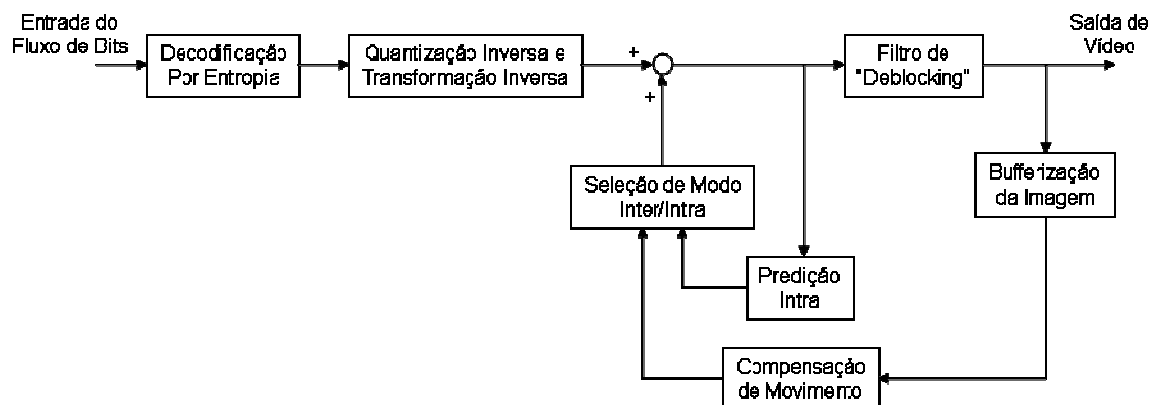


Figura 11 - Diagrama de blocos do decodificador H.264/AVC.

2.14 As Camadas VCL e NAL do Padrão H.264/AVC

O padrão H.264/AVC consiste de duas camadas: *Video Coding Layer* (VCL) e *Network Abstraction Layer* (NAL). A camada NAL abstrai e adapta os dados gerados pela VCL de forma a adequá-los ao transporte através de uma grande variedade de canais de comunicação, bem como armazenamento em diferentes meios. Procurando facilitar a integração com o canal de comunicação, a NAL especifica tanto os formatos de fluxo de bytes quanto de pacotes. O formato de pacotes é usado em aplicações baseadas no uso combinado dos protocolos RTP/UDP/IP [1].

Na camada VCL está o mecanismo central de compressão, e nela se encontram os níveis sintáticos comumente conhecidos como níveis de bloco, macrobloco e fatia. Foi projetada para ser tão independente da rede quanto possível [1].

2.15 Detalhamento da camada NAL

Alguns dos principais elementos que formam a NAL são:

- a) Unidades NAL: os dados codificados são organizados em unidades NAL, que são simplesmente pacotes que contêm um número inteiro de bytes. O primeiro byte de cada unidade NAL é um byte de cabeçalho que contém uma indicação do tipo de dado contido na unidade, e os demais bytes contêm a carga útil do tipo indicado no cabeçalho. Existem duas classes de unidades NAL, chamadas “VCL” e “não-VCL”. As unidades “VCL” contêm os dados que representam os valores das amostras das imagens de vídeo. Já as unidades “não-VCL” contêm outras informações relevantes tais como conjuntos de parâmetros (dados de cabeçalho importantes que podem se aplicar a um grande número de unidades NAL “VCL”) e informações suplementares

de temporização e de melhoria geral na utilização do sinal de vídeo decodificado mas que não são necessárias para a decodificação dos valores das amostras das imagens de vídeo.

- b) Conjuntos de parâmetros: um conjunto de parâmetros contém informações de cabeçalho importantes que podem se aplicar a um grande número de unidades NAL. Existem dois tipos de conjuntos de parâmetros: conjuntos de parâmetros de seqüência, que se aplicam a uma série consecutiva de imagens de vídeo codificadas; conjuntos de parâmetros de figura, que se aplicam à decodificação de uma ou mais imagens individuais.
- c) Unidades de acesso: o conjunto de unidades “VCL” e “não-VCL” que está associado a uma única imagem decodificada é chamado de unidade de acesso. A unidade de acesso contém todos os macroblocos da imagem, possivelmente algumas aproximações redundantes de algumas partes da imagem para propósitos de resiliência a erros (chamadas de fatias redundantes) e outras informações suplementares associadas à imagem.

2.16 Detalhamento da camada VCL

Uma seqüência de vídeo codificado consiste de uma seqüência de imagens codificadas. Cada imagem é particionada em macroblocos. No formato 4:2:0, cada macrobloco é composto de uma matriz de amostras luma 16x16 e duas matrizes de amostras chroma 8x8, conforme mostrado na Figura 12. No formato 4:2:2, as matrizes de amostras chroma são 8x16, e no formato 4:4:4, as matrizes são todas 16x16.

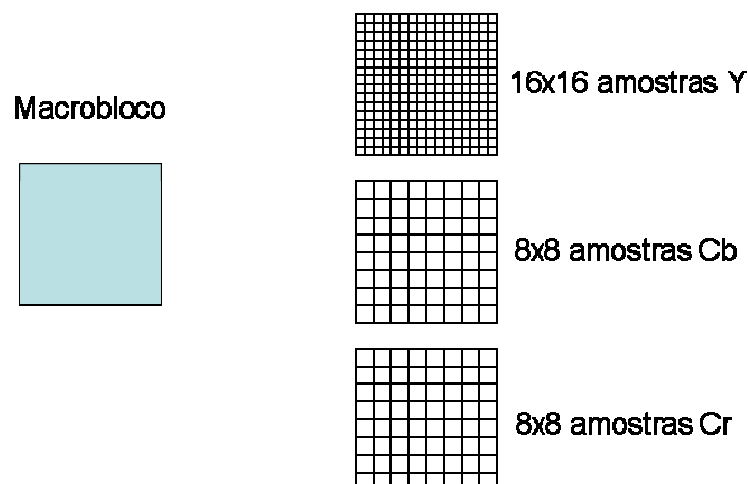


Figura 12 - Macrobloco e amostras Y, Cb e Cr no formato 4:2:0.

Macroblocos são os blocos fundamentais sobre os quais o processo de decodificação é especificado. As amostras de luma e chroma de um macrobloco são preditas – tanto espacial quanto temporalmente – e o residual de predição é transmitido usando-se codificação de transformada. Cada componente de cor do residual é sub-dividido em blocos, cada bloco é transformado usando-se transformação de número inteiro, e os coeficientes da transformação são quantizados e codificados por entropia.

Os macroblocos de uma imagem são organizados em fatias, que representam regiões de uma dada imagem, e que podem ser decodificadas independentemente. Cada fatia é uma seqüência de macroblocos que é processada no sentido do canto superior esquerdo ao canto inferior direito (*raster scan*), apesar de que não precisam necessariamente serem consecutivas no processo de *scan*. Uma imagem, ou quadro, pode conter uma ou mais fatias, como no exemplo da Figura 13 abaixo. Cada fatia é auto-suficiente, isto é, pode ser decodificada sem a necessidade de dados de outras fatias da imagem.

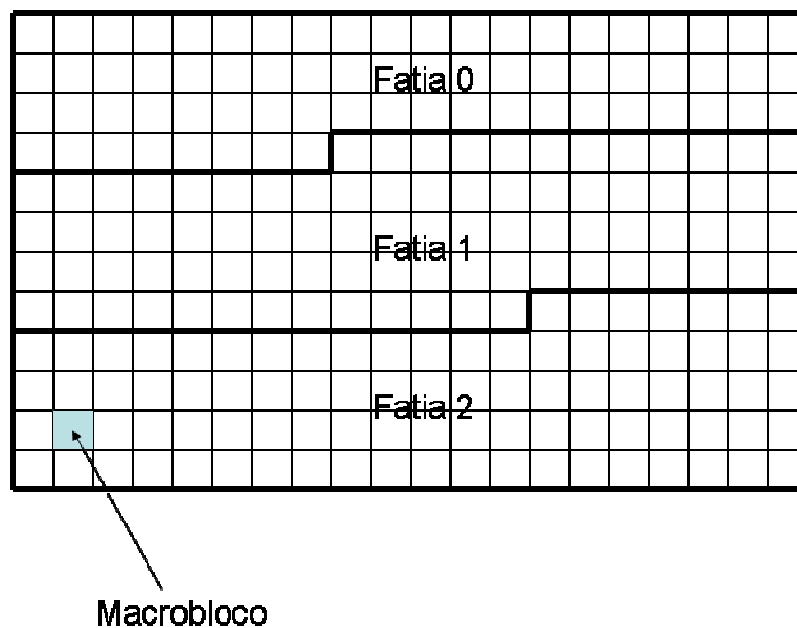


Figura 13 - Imagem de uma seqüência de vídeo dividida em macroblocos e fatias.

Tipos de fatias:

- i) Fatia I: fatia em que todos os macroblocos são codificados usando predição “Intra”.
- ii) Fatia P: fatia codificada com predição “Inter” unidirecional (diferença bloco-a-bloco do último quadro).

- iii) Fatia B: fatia codificada com predição “Inter” bi-direcional (diferenças entre último e próximo quadro, combinadas usando-se média ponderada).
- iv) Fatias SP e SI: as fatias de comutação (*switching*) entre fluxos de bits foram introduzidas no padrão H.264 e servem para facilitar a comutação entre fluxos de bits que correspondem, por exemplo, à mesma sequência de vídeo codificada com taxas de bits diferentes.

2.17 Variabilidade de Tráfego

Auwer *et al.* [18] mostram que a variabilidade do tráfego gerado pelo padrão H.264/MPEG-4 AVC é significativamente maior do que aquela gerada pelos padrões MPEG-2 e MPEG-4 Part 2. Enquanto o coeficiente de variação de tamanho de quadro (definido como o desvio padrão dividido pela média) atinge níveis acima de 2,4 no caso do padrão H.264/MPEG-4 AVC, o mesmo coeficiente não ultrapassa 1,5 no caso do MPEG-4 Part 2. Tal nível acima de 1,5 é inédito.

Auwer *et al.* [19] também mostram que a maior variabilidade de tráfego gerada pelo codificador H.264/MPEG-4 AVC aumenta significativamente as perdas de quadros, se comparado ao padrão MPEG-4 Part 2, na transmissão de fluxos de vídeo através de um enlace do tipo gargalo.

Tais conclusões também serviram de motivação para este trabalho, onde foram aplicadas técnicas de AQM sobre vídeos H.264/MPEG-4 AVC buscando uma possível minimização de perdas de pacotes, em comparação com a técnica *droptail* tradicional.

3. APRESENTAÇÃO DA METODOLOGIA

A metodologia escolhida para este trabalho foi a simulação computacional de uma rede IP que transmite um fluxo de vídeo H.264/MPEG-4 AVC em tempo real, disputando os recursos da rede com tráfego TCP, o tipo mais comumente encontrado na Internet.

Os vídeos foram codificados *offline* usando o software de referência JM versão 13.0 [20]. As simulações foram implementadas no software de simulação de redes Network Simulator versão 2 (NS2), utilizando código adicional para a simulação do envio de vídeo a partir de arquivos *trace* desenvolvido por Chih-Heng Ke *et. al* [21], e a análise de dados foi feita *offline* a partir dos arquivos *trace* gerados pelo NS2.

3.1 Topologia e Configuração da Rede Simulada

Sobre uma topologia *dumb-bell*, onde os roteadores R0 e R1 formam o enlace principal, foram criados enlaces secundários unindo o roteador R0 e as fontes de tráfego, bem como o roteador R1 e os destinos do tráfego, conforme mostra a Figura 14. Os enlaces secundários usados pelas fontes de tráfego TCP foram configurados com taxa de 10 Mbps e tempo de transmissão variável, enquanto que o enlace usado pela fonte de vídeo foi configurado com taxa de 10 Mbps e tempo de propagação de 2 ms. O enlace principal foi configurado com taxa de 2 Mbps e tempo de propagação de 5 ms, caracterizando o gargalo de transmissão. Essa topologia foi escolhida porque tem sido tradicionalmente utilizada na avaliação de desempenho de técnicas de AQM [22] [23].

Conforme mostra a Figura 15, cada rodada de simulação tem duração de 50 segundos, onde os fluxos TCP são disparados em tempos que variam entre 0,1 e 0,7 segundos. O vídeo é disparado em $t = 5$ segundos – tempo suficiente para que os transientes iniciais dos mecanismos de controle de fluxo TCP tenham desaparecido – e interrompido em $t = 49$ segundos, ou seja, cada sequência de vídeo de 10 segundos é repetida 4,4 vezes.

Inicialmente foi criado um conjunto de 5 fluxos TCP no mesmo sentido que o fluxo de vídeo, com diferentes tamanhos de pacotes (120, 300, 500, 900 e 1200 bytes), procurando sobrecarregar os algoritmos de gerenciamento de filas. Esse conjunto de 5 fluxos foi então multiplicado por 2, 3, 6 e 12, chegando a 60 fluxos TCP concorrentes ao final das simulações.

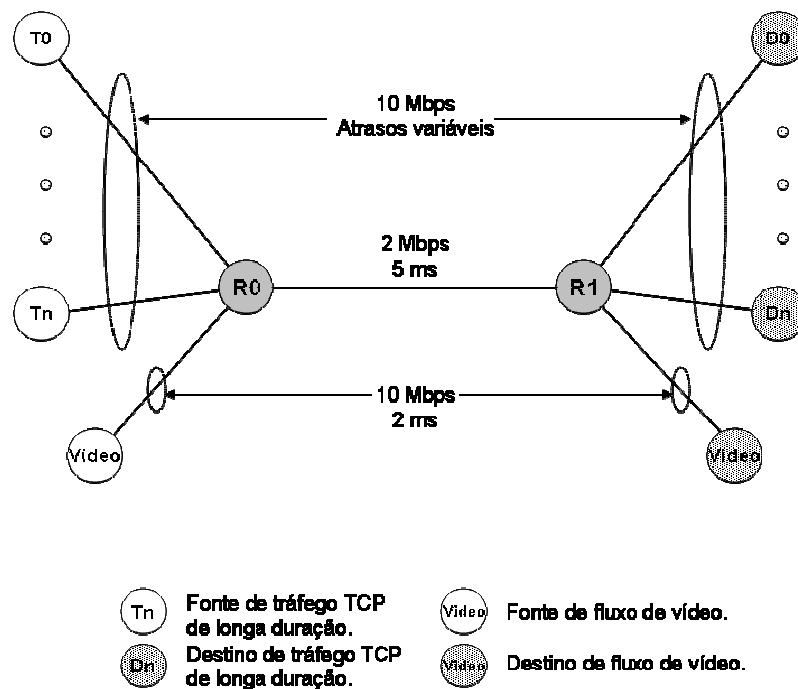


Figura 14 - Topologia da rede simulada.

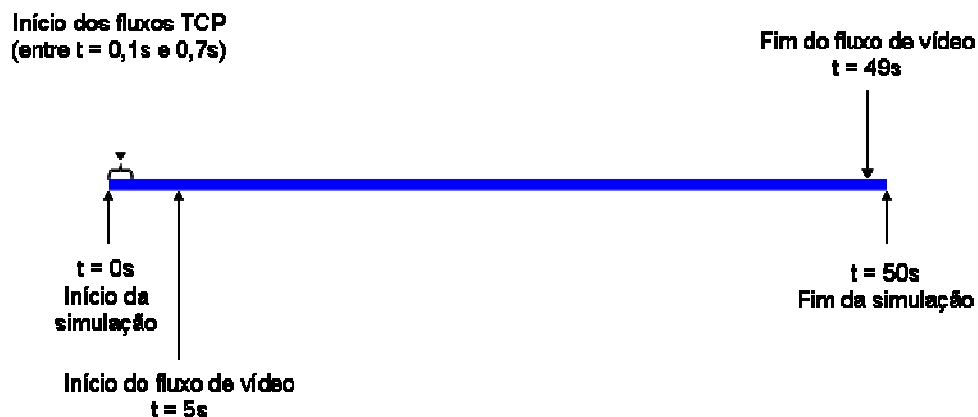


Figura 15 – Tempos de início e fim dos fluxos simulados.

Os dois roteadores que formam o enlace principal foram simulados dentro de dois cenários distintos no que diz respeito ao gerenciamento das filas: aplicando-se a técnica *droptail* tradicional e aplicando-se a técnica ARED. Em ambos os cenários, as filas foram dimensionadas em 120 pacotes. No cenário de operação com a técnica ARED foram especificados dois parâmetros: o atraso médio – configurado em 40ms - e o tamanho médio dos pacotes – calculado a partir da simulação *droptail* para o respectivo número de fluxos TCP concorrentes. Além disso, dentro do cenário ARED foram simulados os dois possíveis modos de operação: calculando o tamanho da fila em bytes e em pacotes.

Em resumo, as simulações compreenderam as combinações de técnicas de gerenciamento de filas, modos de operação e quantidades de fluxos TCP concorrentes conforme a Tabela 1.

Tabela 1 - Combinações de técnicas de gerenciamento de filas, modos de operação e quantidade de fluxos TCP utilizados nas simulações.

Técnica	Tamanho da Fila	Modo de Operação	Atraso Médio	Quant. Fluxos TCP
Droptail	120 pacotes	(não se aplica)	(não se aplica)	5, 10, 15, 30, 60
ARED	120 pacotes	Byte	40ms	5, 10, 15, 30, 60
	120 pacotes	Pacote	40ms	5, 10, 15, 30, 60

3.2 Fluxos de Vídeo Simulados

Os fluxos de vídeo H.264/MPEG-4 AVC foram gerados a partir de seqüências de vídeo bruto no formato CIF 352x288 pixels, 4:2:0 YUV, com 10 segundos de duração e amostrados a 30 quadros por segundo. As seqüências utilizadas foram news_cif.yuv e coastguard_cif.yuv, ambas disponíveis em <http://trace.eas.asu.edu/yuv/index.html> [25].

As figuras 16 e 17 mostram os quadros de número 1, 50, 100, 150 e 200 (de um total de 300 quadros) dos vídeos “News” e “Coastguard”, a fim de demonstrar a diferença em termos de variação de quadros entre as duas seqüências: uma seqüência em que predominam semelhanças entre quadros (“News”), com dois apresentadores de telejornal praticamente estáticos e um fundo que varia muito pouco, e uma seqüência em que predominam diferenças entre quadros (“Coastguard”), com os barcos, o fundo da imagem e a água se movimentando constantemente.



Figura 16 - Quadros da seqüência de vídeo “News”.



Figura 17 - Quadros da seqüência de vídeo “Coastguard”.

Essas duas seqüências foram escolhidas por apresentarem características bastante distintas, tanto em termos da razão de compressão quanto da variabilidade de tráfego. Enquanto *coastguard_cif.yuv* apresenta uma alta taxa de variação de imagens entre quadros, a seqüência *news_cif.yuv* apresenta imagens com baixa taxa de variação. Como resultado, atinge-se uma alta razão de compressão e alta variabilidade de tráfego no caso da seqüência *news_cif.yuv*, e baixa razão de compressão e baixa variabilidade de tráfego no caso da seqüência *coastguard_cif.yuv*. A Tabela 2 lista os dados estatísticos mais relevantes das seqüências de vídeo codificadas.

Tabela 2 - Dados estatísticos das seqüências de vídeo codificadas.

	news_cif	coastguard_cif
Número de quadros	300	300
Razão de compressão	110,54	33,42
Tamanho médio dos quadros (bytes)	1392,54	4614,04
Tamanho do menor quadro (bytes)	70	2219
Tamanho do maior quadro (bytes)	8603	14581
Razão <i>peak-to-mean</i>	6,18	3,16
Tamanho total da seqüência (bytes)	417761	1384214
Taxa média de transmissão (bits/s)	334208,8	1107371,2

A codificação foi feita através do software de referência JM versão 13.0, usando o perfil “High”, voltado a aplicações de alta resolução e que suporta vídeo amostrado em 8 bits por amostra no formato 4:2:0. O GOP (*Group of Pictures* – grupo de imagens) utilizado foi de um quadro intra-codificado (tipo I) para cada 11 quadros inter-codificados (tipo P), sem quadros codificados bi-direcionalmente (tipo B), resultando num GOP fechado IPPPPPPPPPPP. O parâmetro de quantização foi configurado em 28. (Veja Anexo II para maiores detalhes dos parâmetros de codificação.)

O controle de fluxo H.264/MPEG-4 AVC foi desabilitado por duas razões:

- a) a codificação foi feita *offline*;
- b) o objetivo das simulações é avaliar o impacto das diferentes técnicas de gerenciamento de filas quando um fluxo de alta variabilidade está presente na rede.

Dessa forma, a saída do codificador resultou num fluxo de vídeo do tipo VBR (*Variable Bit Rate* – Taxa de Bits Variável), sem qualquer interação entre o codificador e as condições da rede.

Uma vez terminada a codificação, o vídeo em formato “.264” bruto foi importado para *containers* MP4 e preparado para transmissão utilizando a ferramenta MP4Box, disponível em

<http://www.videohelp.com/tools/MP4Box> [26]. Nessa etapa, cada quadro foi segmentado em pacotes de 200 ou 1024 bytes.

Utilizando a ferramenta MP4TRACE (veja seção 3.3 A Estrutura “Evalvid”) foram gerados os arquivos *trace* usados na simulação. Os respectivos vídeos são identificados na seção 4. RESULTADOS como “News 200”, “News 1024”, “Coastguard 200” e “Coastguard 1024”.

A partir dos arquivos *trace* de saída do NS2 foram calculadas as perdas de pacotes de vídeo, percentuais de quadros de vídeo danificados, atrasos, *jitter* e PSNR dos vídeos resultantes das simulações (veja seção 4. RESULTADOS).

3.3 A Estrutura “Evalvid”

“Evalvid” é o nome de uma estrutura formada por um conjunto de ferramentas para avaliação da qualidade de fluxos de vídeo transmitidos em redes reais ou simuladas [27]. O que torna a estrutura “Evalvid” atraente é a relativa simplicidade da interação entre os arquivos de *trace* usados na avaliação da qualidade de transmissão: são necessários apenas os arquivos de *trace* de entrada (representando o fluxo de vídeo original) e de saída da rede (representando o fluxo de vídeo corrompido). Dessa forma, a rede pode ser encarada como uma “caixa preta” que provoca perdas, atrasos e um possível reordenamento dos pacotes. Assim como a “caixa” que representa a rede, as demais “caixas” que formam a estrutura também podem ser substituídas. Essa característica é particularmente interessante para o caso da “caixa” que representa o codificador de vídeo.

Através da ferramenta MP4TRACE, as seqüências de vídeo codificadas e preparadas para transmissão foram usadas como entrada para a geração de um arquivo de texto do tipo *trace*, que contém todas as informações necessárias para que o NS2 crie um fluxo simulado que se comporta exatamente como o fluxo codificado original em termos de número de seqüência de quadros, tipo de quadro, quantidade e tamanho dos segmentos de quadros (pacotes) enviados, e temporização de transmissão [24]. Os valores de PSNR foram calculados através da ferramenta PSNR a partir dos vídeos reconstruídos.

As ferramentas foram usadas conforme descrito em [27] e [21]. A Figura 18 mostra o diagrama de blocos da estrutura usada na simulação da rede e avaliação dos fluxos de vídeo codificados.

Fonte de vídeo

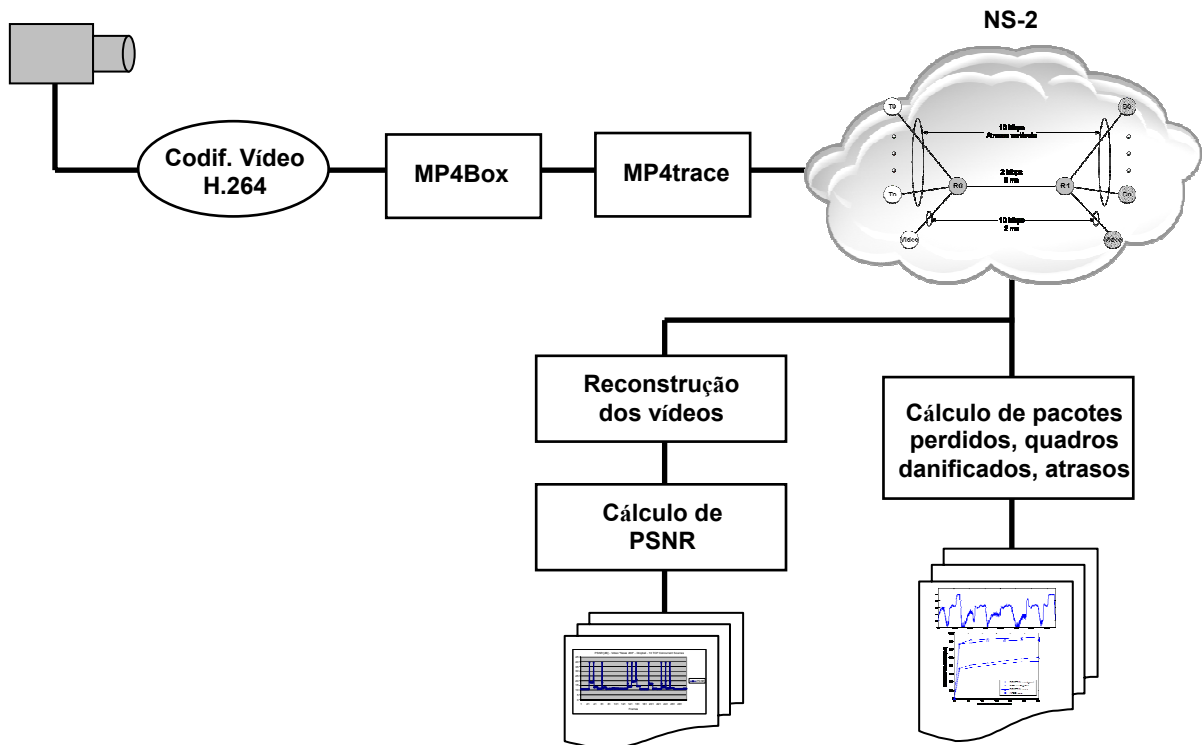


Figura 18 - Estrutura usada na codificação, simulação e avaliação dos fluxos de vídeo.

4. RESULTADOS

Conforme mencionado anteriormente, a pesquisa sobre o impacto da técnica ARED em fluxos de vídeo H.264/MPEG-4 AVC mostrada neste trabalho teve como principal motivação os bons resultados obtidos com a referida técnica em fluxos VoIP simulados em cenários semelhantes. Sendo assim, primeiramente serão descritos resumidamente que resultados foram esses, e em seguida serão apresentados os resultados obtidos com os fluxos de vídeo objeto deste trabalho.

4.1 Impacto da Técnica ARED em Fluxos VoIP

Vitalio *et al.* [4] concluíram que “a técnica ARED operando em modo byte proporciona um excelente tratamento aos pacotes de voz mesmo sob condições de tráfego pesado, chegando a uma qualidade percebida próxima daquela oferecida pelo sistema de telefonia convencional” e que “não houve variação na qualidade de voz percebida quando operando em modo pacote”. Os resultados que levaram a tais conclusões estão expostos abaixo, com a autorização dos autores.

A Tabela 3 resume as medidas para duas chamadas VoIP que disputam um enlace gargalo com 50 fontes de tráfego TCP de longa duração e tráfego web. Quando é usada a técnica *droptail*, os resultados indicam que quase metade dos usuários estariam insatisfeitos com a qualidade da ligação, com base na medida de MOS_{cq} (*Mean Opinion Score for Conversational Quality*) obtida. Perdas de pacotes em rajadas, atrasos e variações no atraso provocados por flutuações próximas do final da fila provocam uma degradação bastante significativa na qualidade da chamada. A Figura 19 mostra o atraso de rede altamente variável dos pacotes de voz de uma das chamadas. Por outro lado, altos níveis de ocupação da fila resultam em utilização total do enlace.

Já a partir da aplicação da técnica ARED (em modo byte, pois pretende-se favorecer os pacotes de voz), o atraso de rede teve uma redução significativa. A Figura 20 mostra que o atraso de enfileiramento ficou muito próximo dos 40 ms configurados como valor alvo. Enquanto a redução no atraso era esperada, a dramática redução nas perdas foi surpreendente (veja Tabela 3). Isso significa que a técnica ARED é capaz de estabilizar o tamanho da fila sem impor uma perda de pacotes agressiva, ao menos para pacotes de voz. Como resultado, o MOS_{cq} ultrapassou 4,20 pontos, similar à qualidade de um sistema de telefonia convencional. O preço pago por isso é uma diminuição na utilização do enlace, que baixou a aproximadamente 95%.

Tabela 3 - Medidas de qualidade de chamadas de voz obtidas por Vitalio *et al.* [4] (reproduzida com autorização dos autores).

Metric	Call No. 1		Call No. 2	
	Droptail	ARED	Droptail	ARED
loss rate (%)	6.26	1.00	6.04	0.91
discard rate (%)	0.00	0.00	0.00	0.00
burst density (%)	30.95	19.57	31.06	17.62
gap density (%)	0.71	0.75	0.56	0.69
burst duration (ms)	142.33	106.82	143.00	119.47
gap duration (ms)	630.81	7723.91	650.57	8886.5
m2e delay (ms)	354.47	205.2	363.25	202.35
R factor	50.24	85.30	50.05	85.95
MOS _{CQ}	2.59	4.21	2.58	4.23

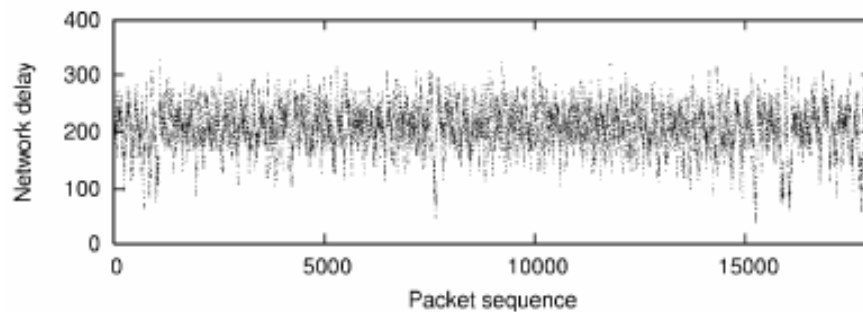


Figura 19 - Atraso de rede dos pacotes de voz sob a técnica *droptail* obtido por Vitalio *et al.* [4] (reproduzida com autorização dos autores).

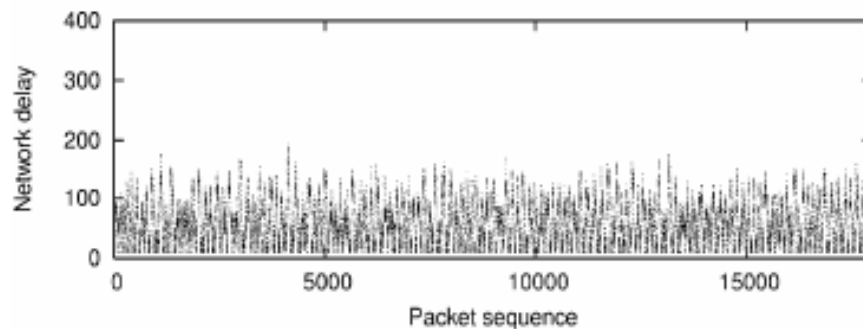


Figura 20 - Atraso de rede dos pacotes de voz sob a técnica ARED obtido por Vitalio *et al.* [4] (reproduzida com autorização dos autores).

Outras simulações revelaram que a técnica ARED exibiu um desempenho excepcional sob diferentes cargas de rede. A Figura 21 mostra o MOS obtido quando o número de fluxos TCP de longa duração variou de 10 a 100. Em todos os casos, o valor do MOS ficou acima dos 4,03 pontos, o limite inferior que atesta satisfação do usuário.

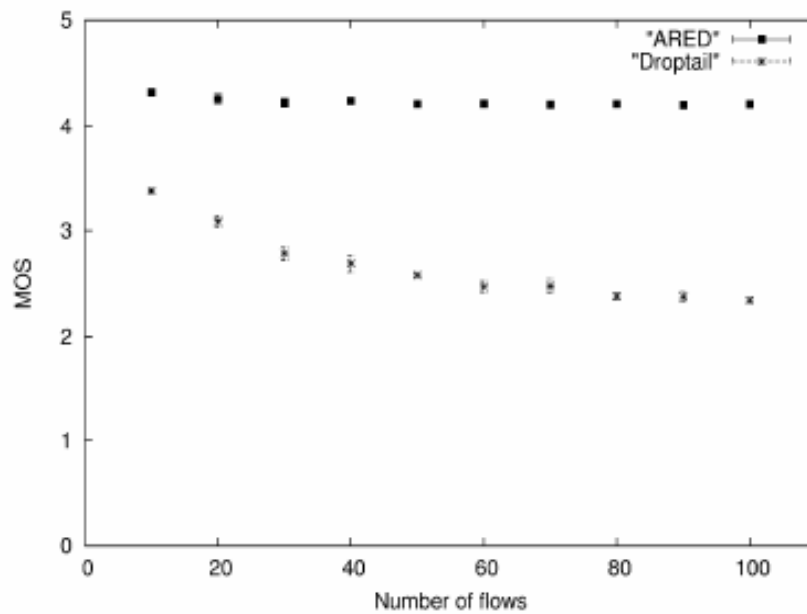


Figura 21 - MOS para qualidade conversacional, comparando as técnicas *droptail* e ARED, obtido por Vitalio *et al.* [4] (reproduzida com autorização dos autores).

Em resumo, Vitalio *et al.* [4] concluíram que, com base nas simulações efetuadas, o uso de esquemas de gerenciamento ativo de filas como ARED melhora significativamente a qualidade percebida em chamadas de voz que atravessam enlaces congestionados.

4.2 Perdas de Informação de Vídeo

As perdas de informação de vídeo foram contabilizadas como porcentagens das quantidades totais de pacotes e quadros de vídeo contidos numa simulação. Foram considerados danificados todos os quadros que sofreram perdas de um ou mais pacotes.

4.2.1 Técnica Droptail

A Figura 22 mostra as porcentagens de pacotes perdidos e quadros danificados do vídeo “News”, e a Figura 23 mostra as porcentagens de pacotes perdidos e quadros danificados do vídeo “Coastguard”, ambas referentes à técnica *droptail*.

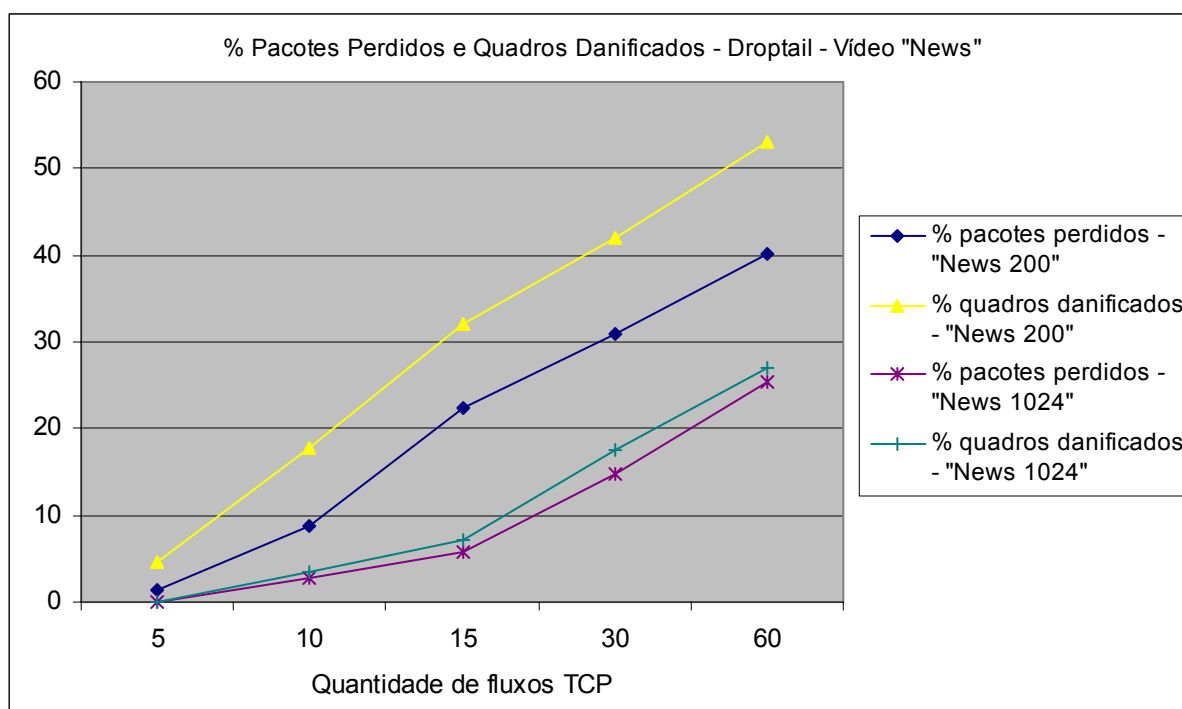


Figura 22 - Vídeo “News” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica *droptail* nos roteadores.

As curvas de perdas de pacotes e quadros danificados são coerentes com os resultados esperados do comportamento de descarte da técnica *droptail*. À medida em que o número de fluxos TCP concorrentes aumenta, também aumenta a porcentagem de pacotes descartados. Também fica claro que um determinado vídeo segmentado em pacotes de 200 bytes está mais sujeito a descartes do que se tivesse sido segmentado em pacotes de 1024 bytes, assim como

existe uma amplificação da porcentagem de perdas de quadros em relação às perdas de pacotes pois os quadros estão distribuídos em uma maior quantidade de pacotes.

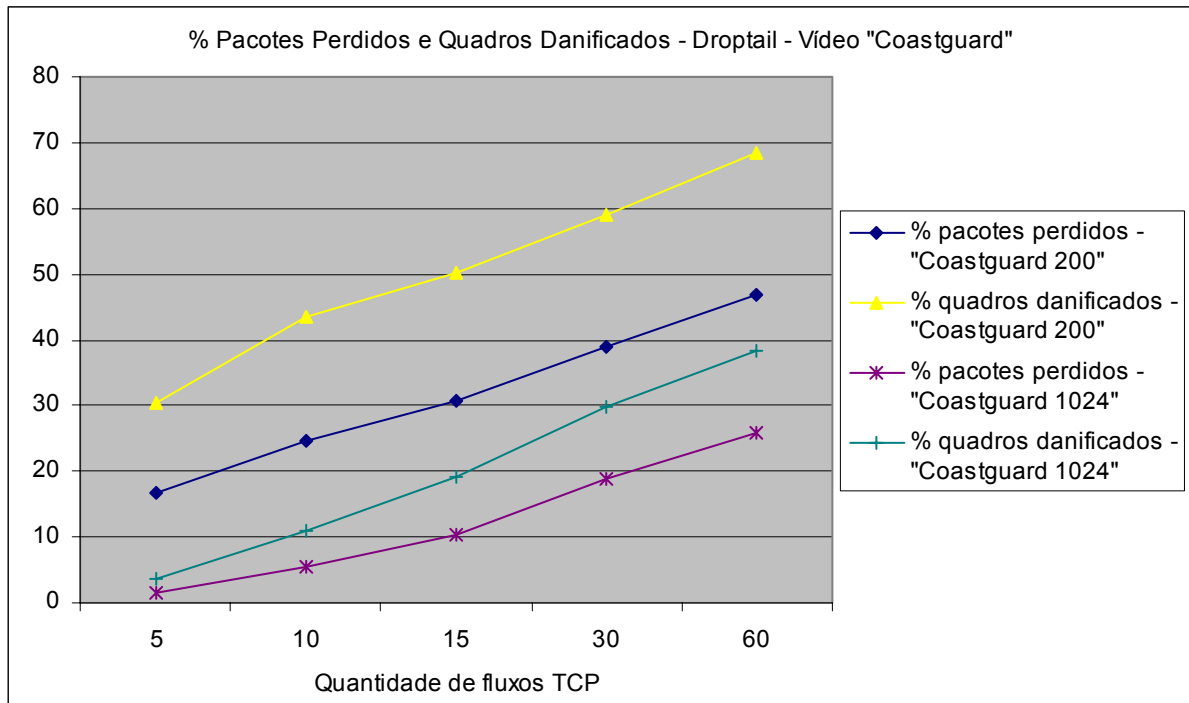


Figura 23 - Vídeo "Coastguard" - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica *droptail* nos roteadores.

No entanto, é interessante frisar que as porcentagens de perdas de pacotes e quadros danificados seguem praticamente a mesma inclinação. Tal efeito pode ser explicado pela característica de perdas em rajadas que a técnica *droptail* provoca. Ao perder pacotes em rajadas, a concentração de perdas num dado instante danifica uma quantidade limitada de quadros de vídeo, afetando uma sequência de pacotes que compõem um único quadro ou uns poucos quadros adjacentes.

4.2.2 Técnica ARED

As figuras 24 a 27 mostram as porcentagens de pacotes perdidos e de quadros danificados para os vídeos "News 200", "News 1024", "Coastguard 200" e "Coastguard 1024" respectivamente, todas referentes à técnica ARED operando nos modos byte e pacote.

Na Figura 24 percebe-se um efeito esperado da técnica ARED operando em modo byte: o favorecimento de pacotes pequenos, se comparado ao modo pacote. Também fica evidente que as inclinações das curvas de perdas de pacotes e quadros danificados são bastante diferentes,

tanto no modo byte quanto no modo pacote, sendo que as curvas de quadros danificados apresentam uma tendência de subida mais rápida. Essa diferença pode ser explicada pelo fato de a técnica ARED diminuir a flutuação do tamanho das filas a partir de descartes aleatórios distribuídos no tempo. Se por um lado tal característica consegue manter o atraso de enfileiramento sob controle e próximo do valor configurado como parâmetro da simulação, por outro faz com que um maior número de quadros seja afetado pelas perdas.

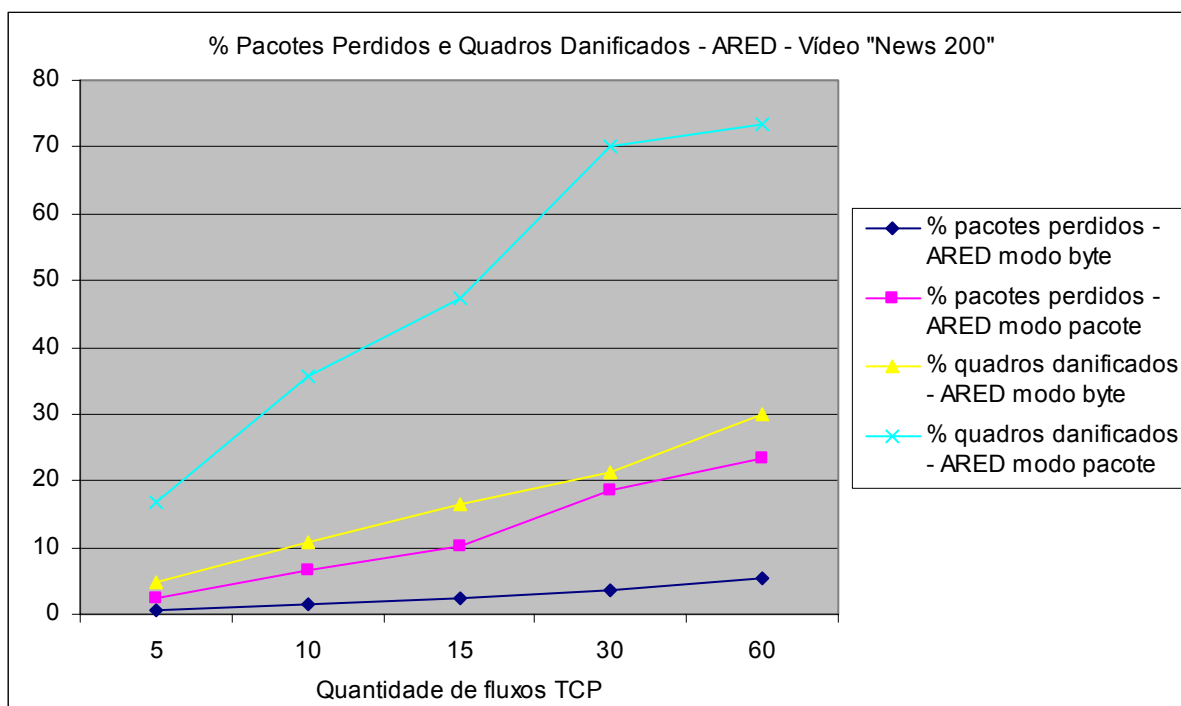


Figura 24 - Vídeo “News 200” - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.

Já no caso do vídeo “News 1024” mostrado na Figura 25 percebe-se que a técnica ARED no modo pacote não proporciona um favorecimento aos pacotes de vídeo de tamanho 1024 bytes tão evidente quanto o favorecimento que o modo byte proporciona aos pacotes de 200 bytes. Além disso, a tendência de subida mais rápida das curvas de quadros danificados, percebida na Figura 24, desaparece. Pode-se explicar tal fenômeno considerando que o tamanho médio dos quadros do vídeo “News” (1392 bytes) permite que grande parte dos quadros caiba em um único pacote de 1024 bytes ou menos, ou seja, os descartes distribuídos no tempo afetam um número menor de quadros se comparado ao vídeo “News 200”.

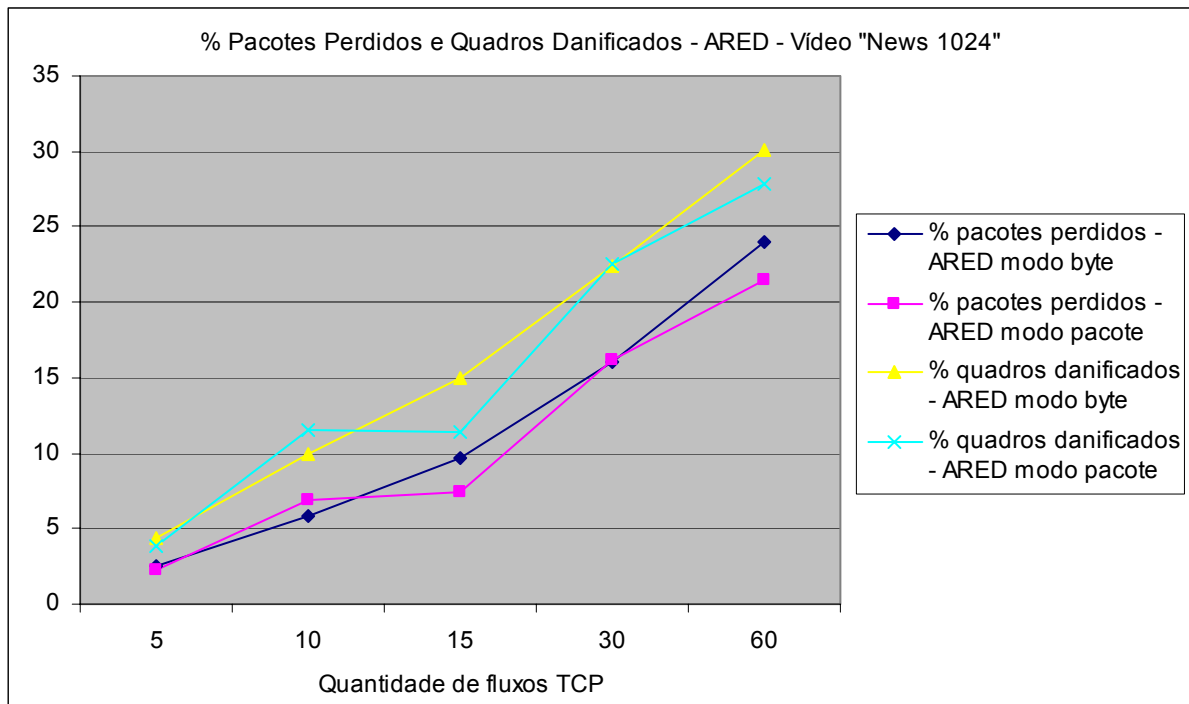


Figura 25 - Vídeo "News 1024" - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.

A Figura 26 mostra o mesmo efeito percebido no vídeo "News 200", porém com uma intensidade maior: uma proporção relativamente baixa de pacotes perdidos danificando uma enorme proporção de quadros. A explicação para o efeito desproporcional é a mesma do vídeo "News 200", já a explicação para a maior intensidade desse efeito é que o maior tamanho médio dos quadros do vídeo "Coastguard" (4614 bytes) distribui a informação referente a um único quadro em aproximadamente 23 pacotes de 200 bytes, em média, ao passo que os quadros do vídeo "News 200" estão distribuídos em aproximadamente 7 pacotes, em média.

A Figura 27 volta a mostrar que a técnica ARED no modo pacote não proporciona um favorecimento aos pacotes de vídeo de tamanho 1024 bytes tão evidente quanto o favorecimento que o modo byte proporciona aos pacotes de 200 bytes, no entanto o maior tamanho médio dos pacotes do vídeo "Coastguard" distribui a informação dos quadros num número de pacotes maior que o vídeo "News", amplificando o prejuízo causado pelas perdas de pacotes, inclusive no caso dos pacotes de 1024 bytes.

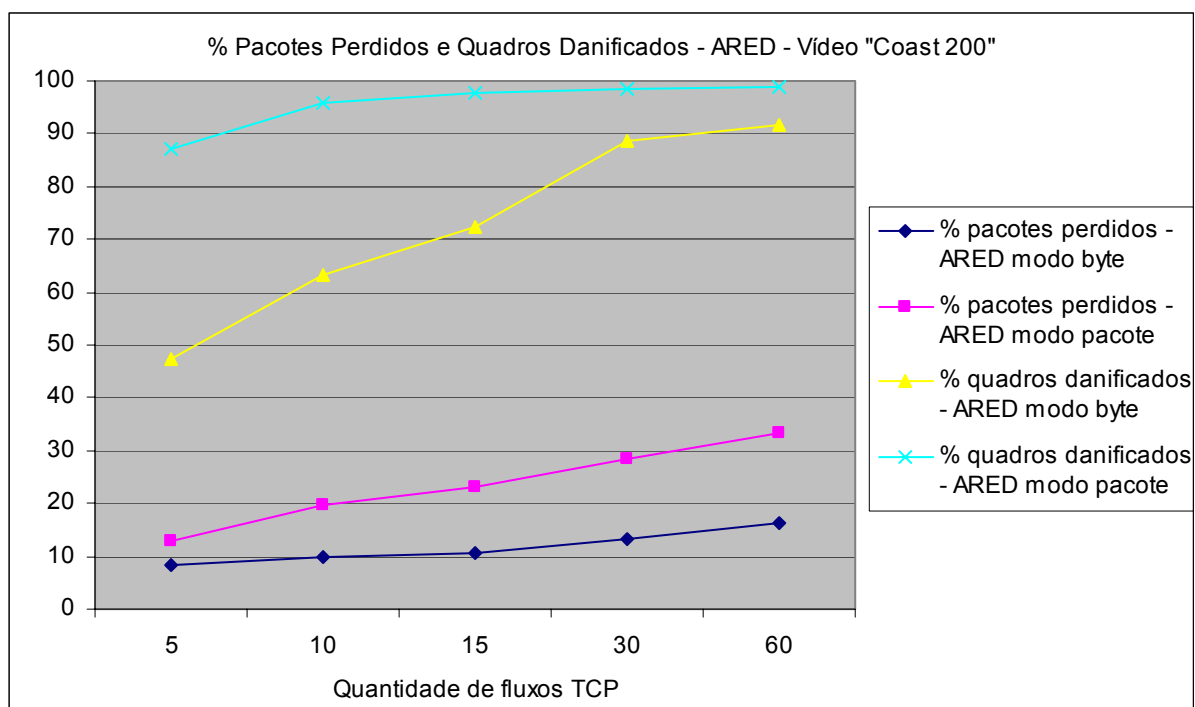


Figura 26 - Vídeo "Coastguard 200" - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.

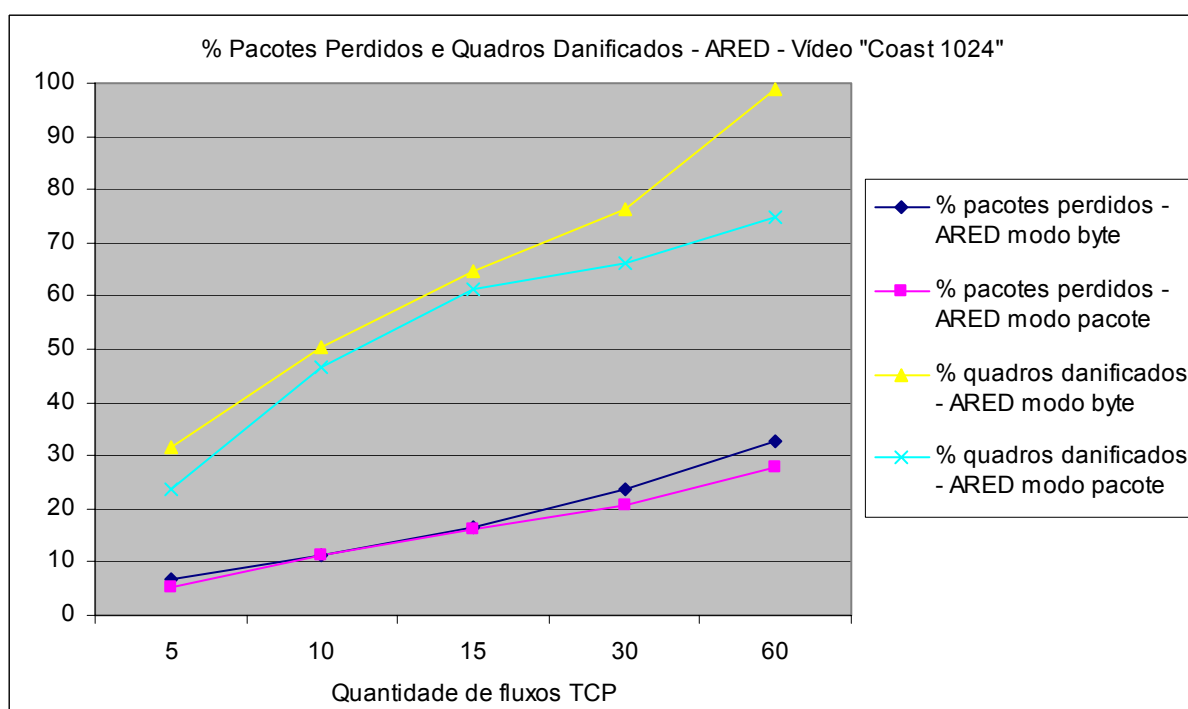


Figura 27 - Vídeo "Coastguard 1024" - porcentagem de pacotes perdidos e quadros danificados utilizando a técnica ARED.

4.2.3 Comparação entre as Técnicas *Droptail* e ARED

As figuras 28 a 31 mostram uma comparação direta entre as técnicas *droptail* e ARED, tanto em modo byte quanto pacote, no que diz respeito à porcentagem de pacotes perdidos. Os resultados obtidos para os vídeos segmentados em pacotes de 200 bytes, que mostram uma porcentagem menor de perdas da técnica ARED em relação à *droptail*, estão de acordo com o comportamento esperado da rede simulada.

Já os resultados obtidos para os vídeos segmentados em pacotes de 1024 bytes surpreendem pela ligeira vantagem obtida pela técnica *droptail* em relação à técnica ARED, seja em modo byte ou pacote.

Tais resultados evidenciam o favorecimento que a técnica ARED, especialmente trabalhando em modo byte, dá aos pacotes de tamanho pequeno (200 bytes), e a indiferença da aplicação dessa técnica dentro das condições simuladas quando os pacotes possuem tamanhos relativamente grandes (1024 bytes).

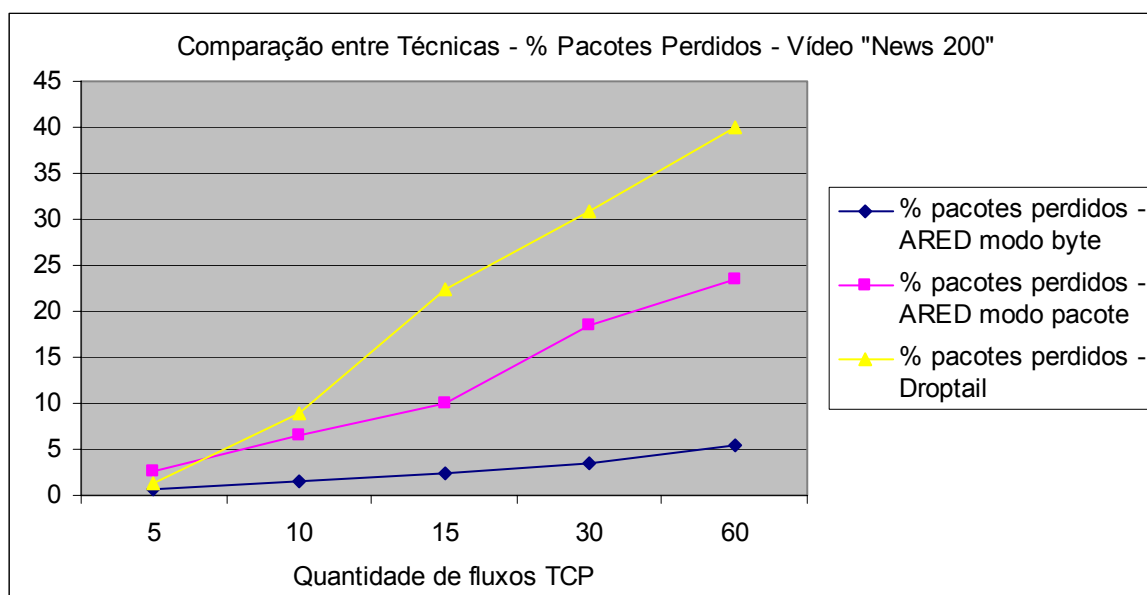


Figura 28 - Vídeo "News 200" - Porcentagem de pacotes perdidos na comparação entre as técnicas *droptail* e ARED.

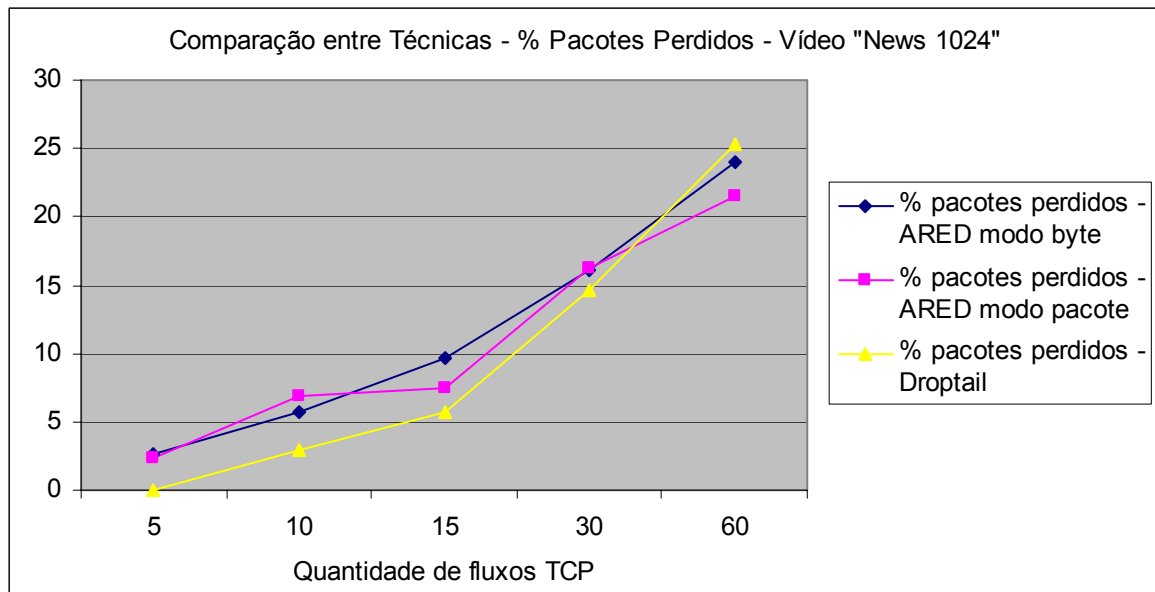


Figura 29 - Vídeo "News 1024" - Porcentagem de pacotes perdidos na comparação entre as técnicas *droptail* e ARED.

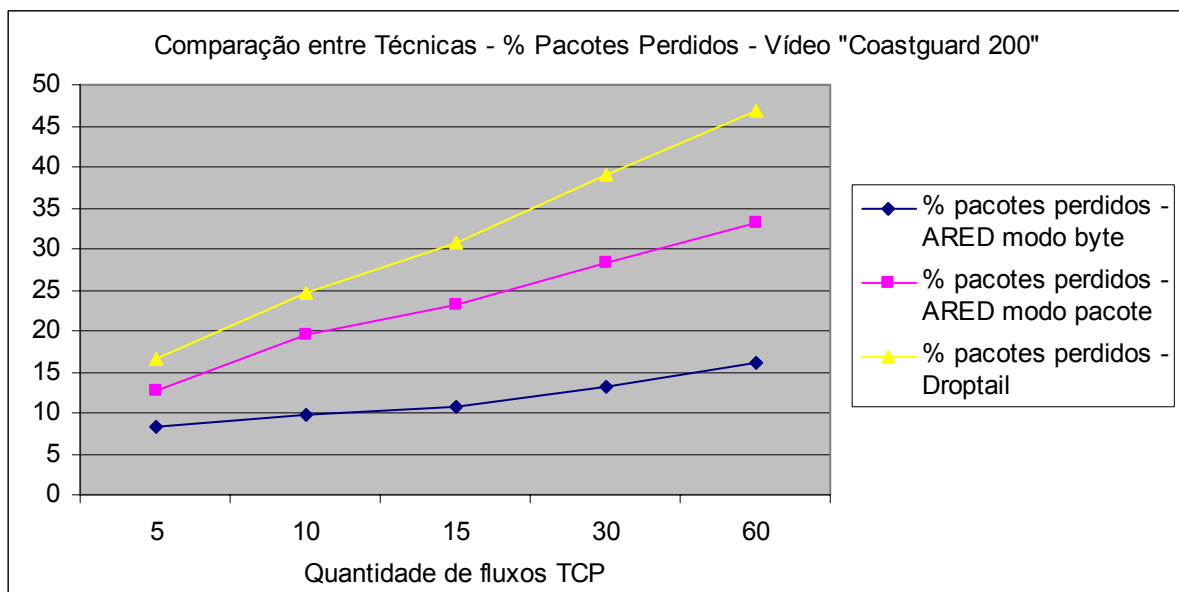


Figura 30 - Vídeo "Coastguard 200" - Porcentagem de pacotes perdidos na comparação entre as técnicas *droptail* e ARED.

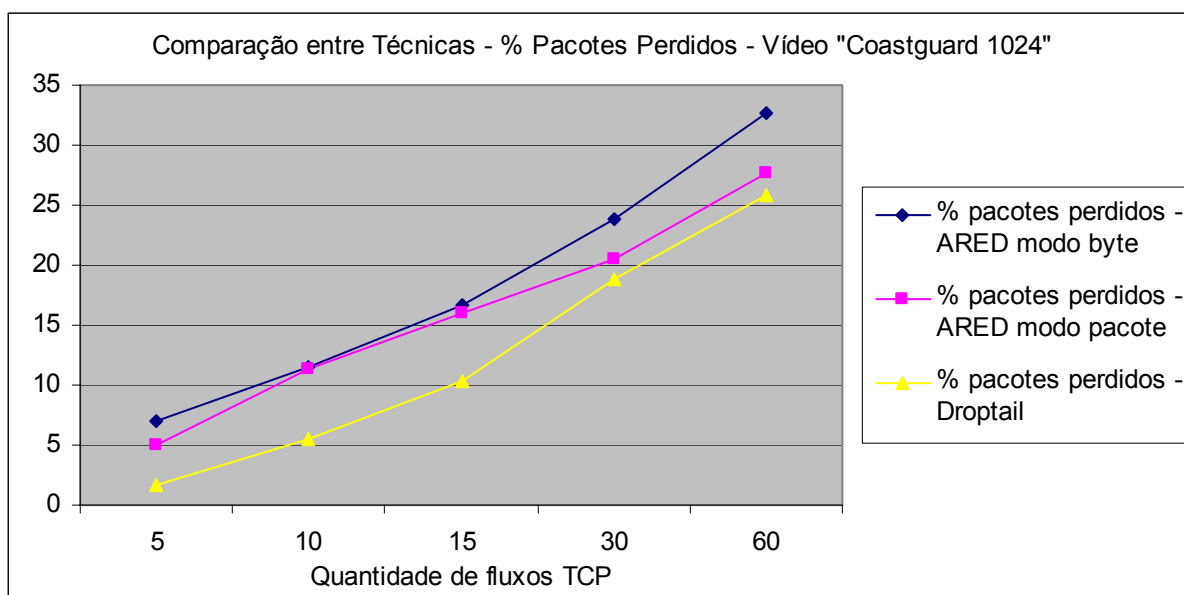


Figura 31 - Vídeo "Coastguard 1024" - Porcentagem de pacotes perdidos na comparação entre as técnicas *droptail* e ARED.

As figuras 32 a 35 mostram uma comparação direta entre as técnicas *droptail* e ARED, tanto em modo byte quanto pacote, no que diz respeito à porcentagem de quadros danificados por perdas de um ou mais pacotes.

Percebe-se que, com exceção de um cenário, em todos os outros a técnica ARED provocou danos em uma porcentagem maior de quadros que a técnica *droptail*, seja em modo byte ou pacote. O único cenário em que a técnica ARED danificou menos quadros que a técnica *droptail* foi com o vídeo "News 200", operando em modo byte, como mostra a Figura 32.

Tais resultados são interessantes pois revelam uma característica nociva da técnica ARED ao transporte de fluxos de vídeo, que fica mascarada pela menor quantidade de perdas de pacotes que ela impõe: apesar de a quantidade de perdas realmente ser menor que a técnica *droptail*, o número de quadros de vídeo danificados é expressivamente maior devido à distribuição dos descartes de pacotes em intervalos regulares de tempo.

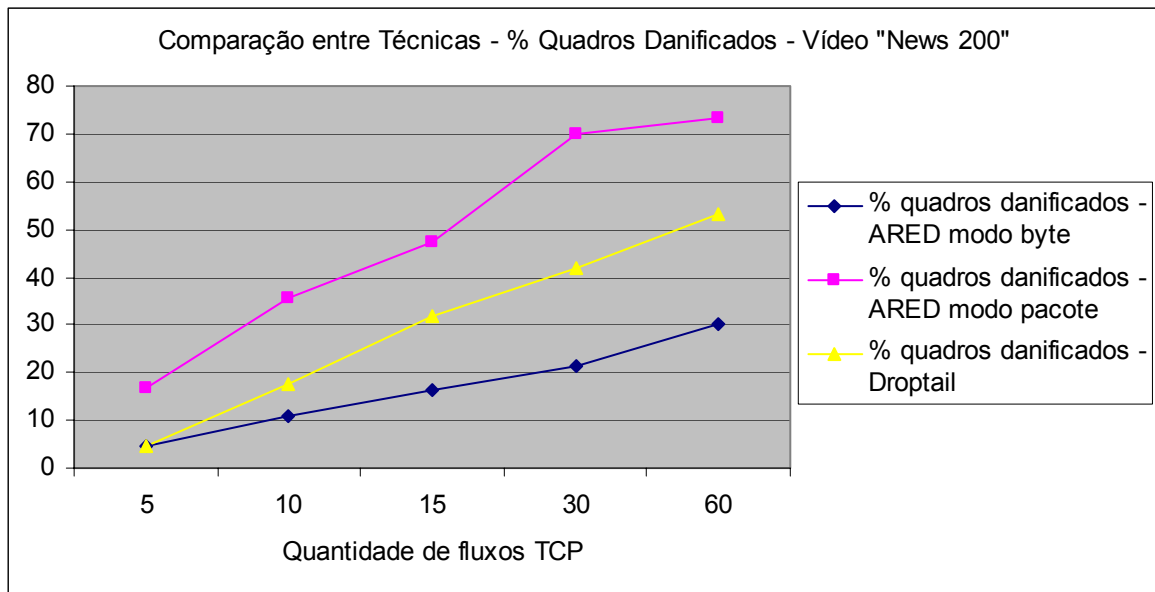


Figura 32 - Vídeo "News 200" - Porcentagem de quadros danificados na comparação entre as técnicas *droptail* e ARED.

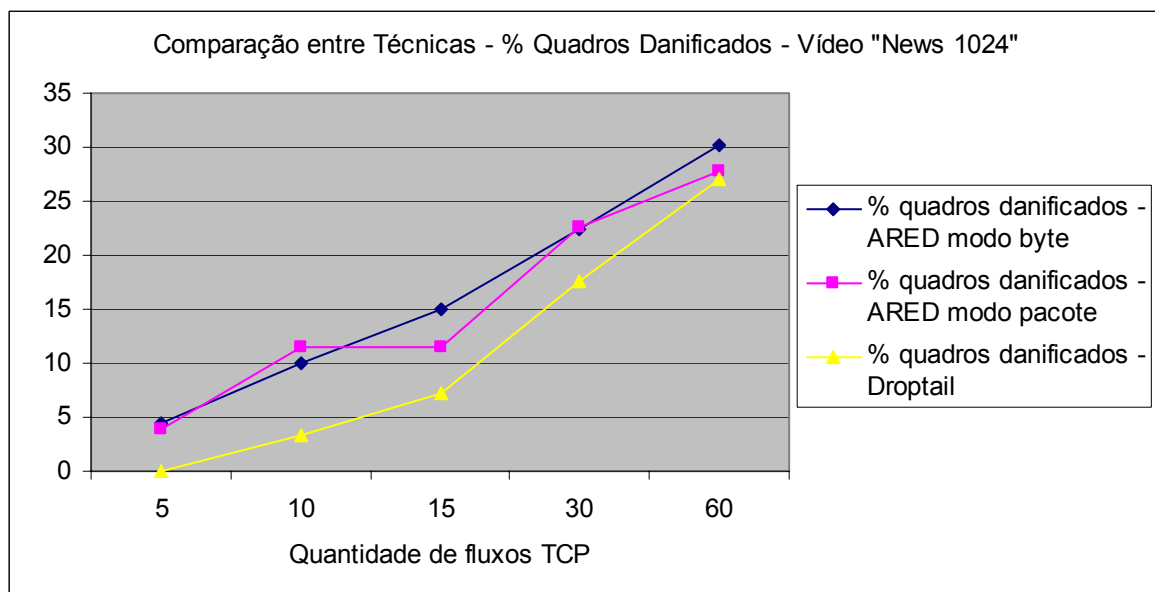


Figura 33 - Vídeo "News 1024" - Porcentagem de quadros danificados na comparação entre as técnicas *droptail* e ARED.

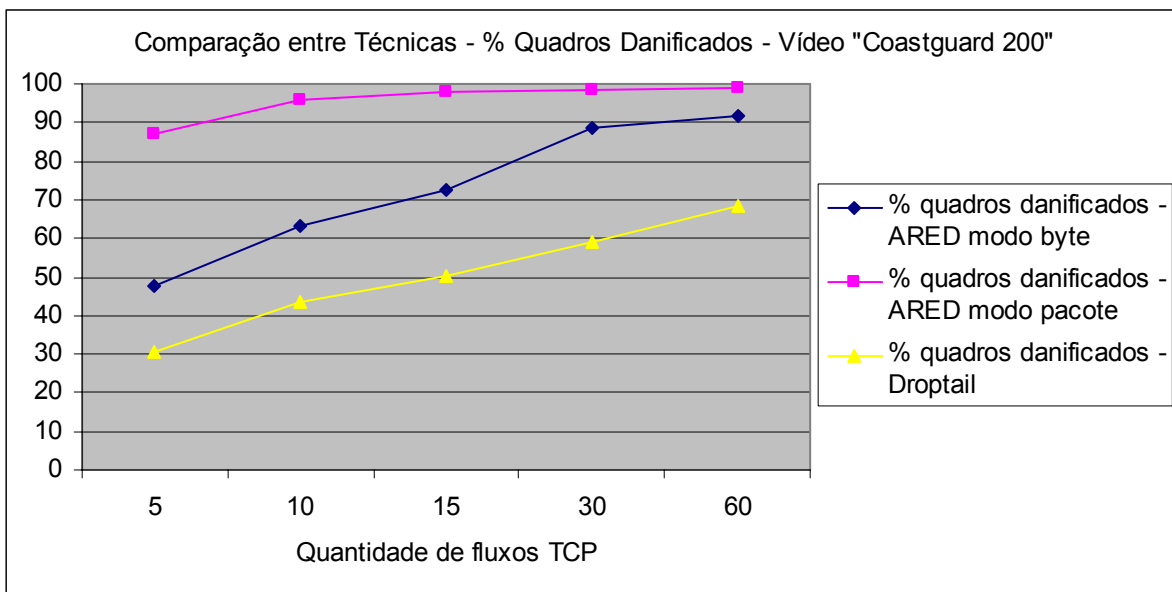


Figura 34 - Vídeo "Coastguard 200" - Porcentagem de quadros danificados na comparação entre as técnicas *droptail* e ARED.

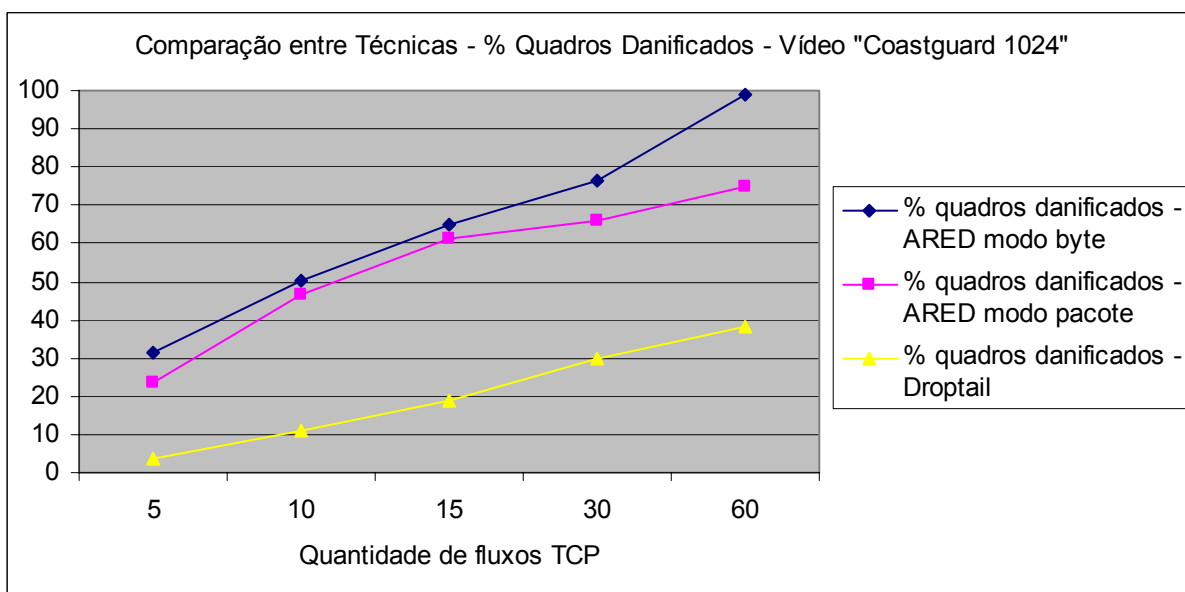


Figura 35 - Vídeo "Coastguard 1024" - Porcentagem de quadros danificados na comparação entre as técnicas *droptail* e ARED.

As figuras 36 a 39 mostram a proporção de quadros do tipo I danificados, de acordo com a técnica aplicada. Considerando que a incapacidade de decodificação de um quadro do tipo I se propaga a todos os quadros do tipo P que pertencem ao mesmo GOP, e levando em conta que o GOP dos fluxos de vídeo simulados neste trabalho é fechado em 12 quadros, a cada quadro do tipo I danificado teríamos um trecho de aproximadamente 400ms de vídeo comprometido.

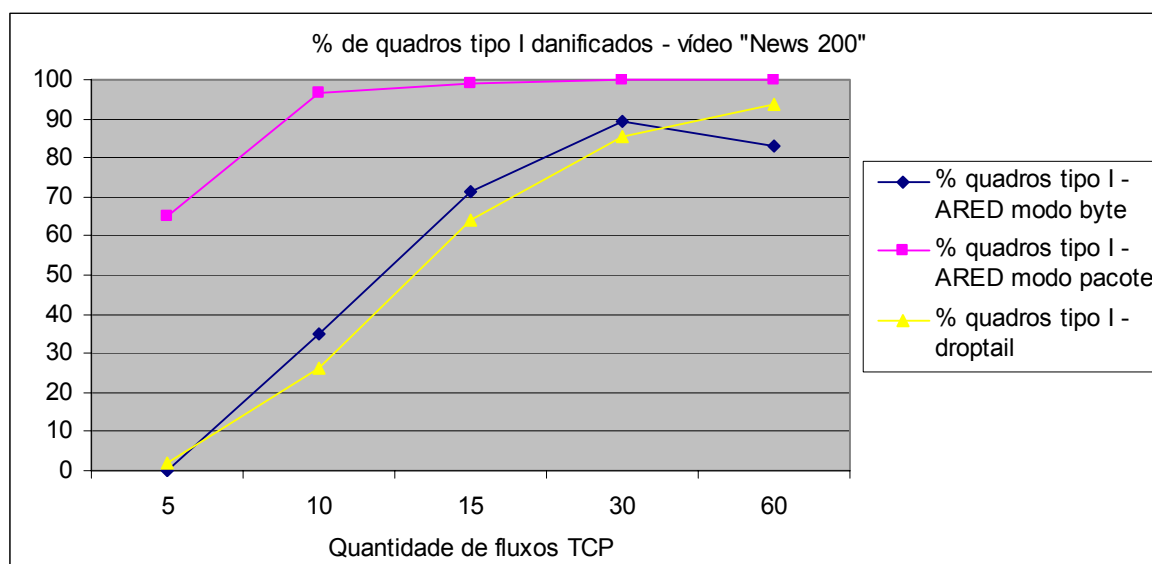


Figura 36 - Vídeo "News 200" - Porcentagem de quadros tipo I danificados na comparação entre as técnicas *droptail* e ARED.

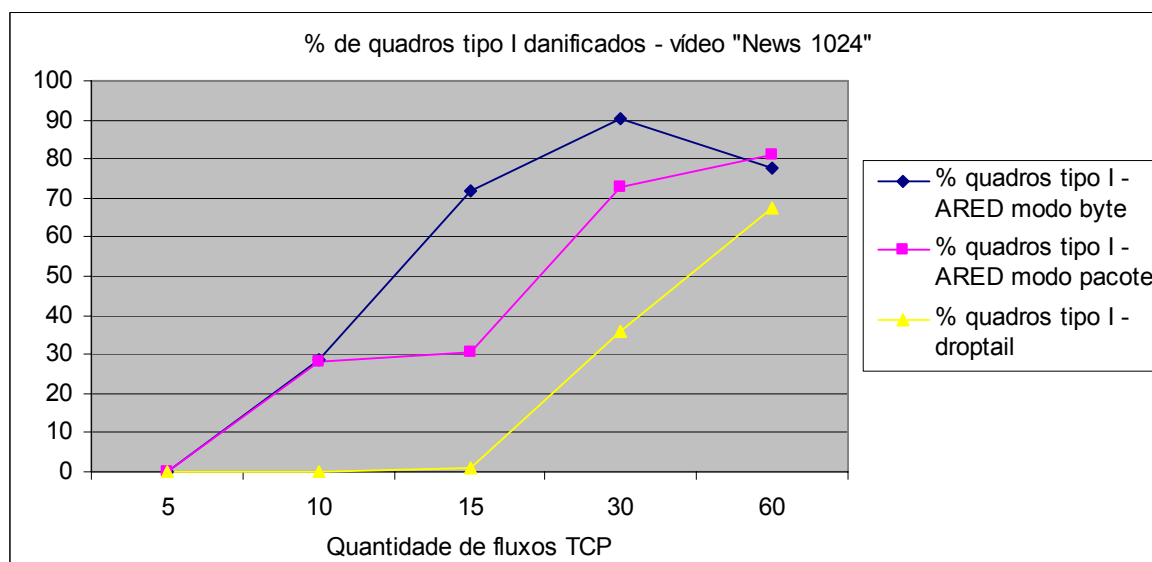


Figura 37 - Vídeo "News 1024" - Porcentagem de quadros tipo I danificados na comparação entre as técnicas *droptail* e ARED.

Novamente percebe-se o efeito nocivo dos descartes distribuídos no tempo que a técnica ARED impõe, afetando uma proporção maior de quadros tipo I que a técnica *droptail* em praticamente todos os cenários simulados. Já a Figura 38, referente ao vídeo “Coastguard 200”, mostra uma situação desastrosa para ambas as técnicas, onde praticamente todos os quadros tipo I foram afetados.

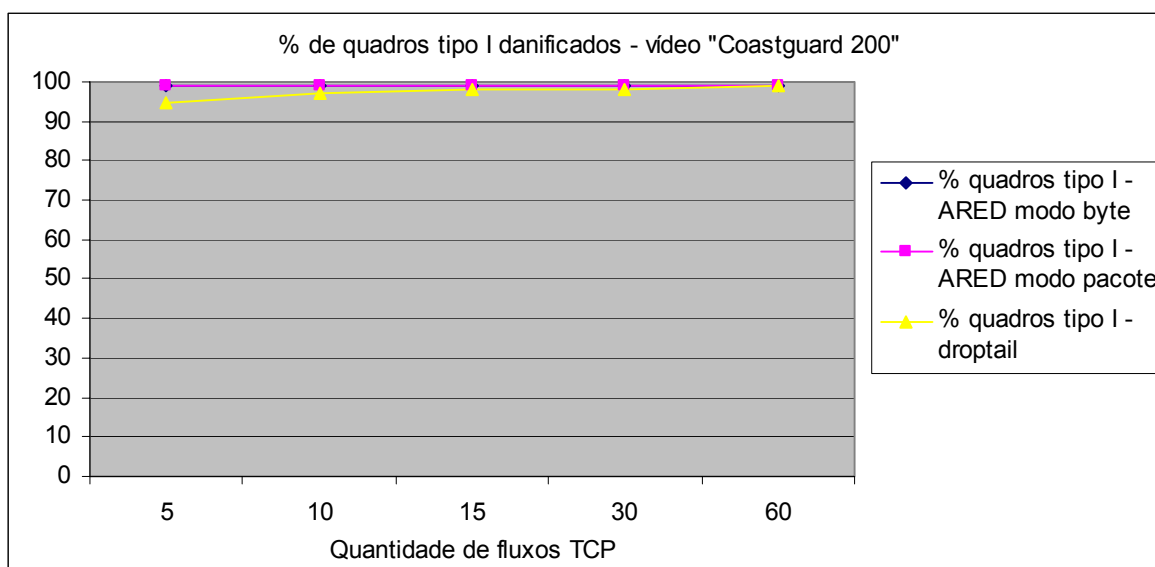


Figura 38 - Vídeo “Coastguard 200” - Porcentagem de quadros tipo I danificados na comparação entre as técnicas *droptail* e ARED.

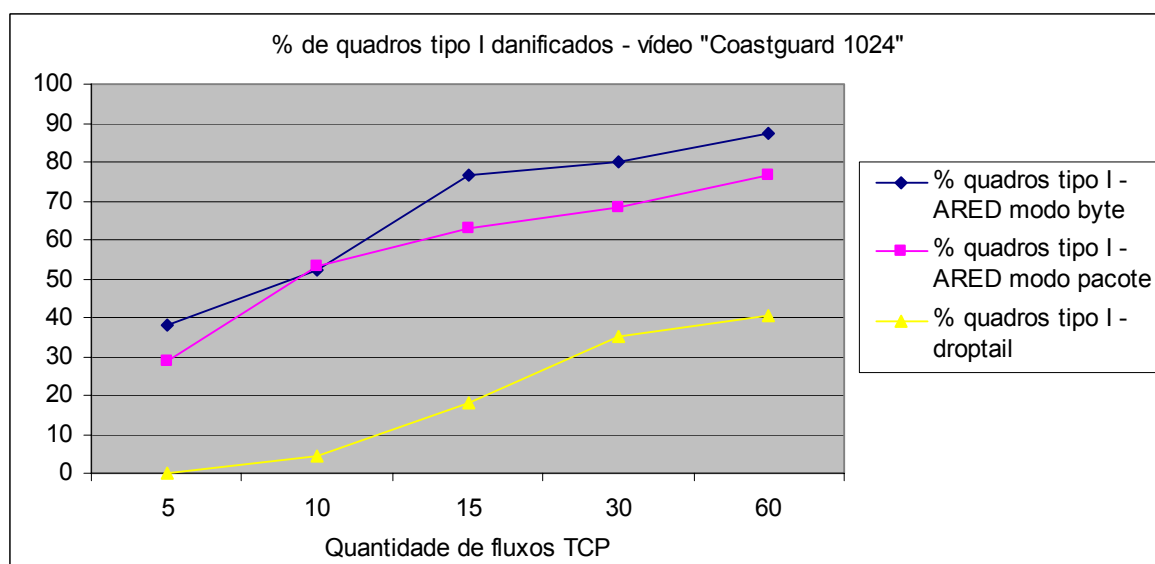


Figura 39 - Vídeo “Coastguard 1024” - Porcentagem de quadros tipo I danificados na comparação entre as técnicas *droptail* e ARED.

4.3 Atrasos Médios, Desvio Padrão dos Atrasos e Jitter

As figuras 40 a 47 mostram o atraso médio de rede, desvio padrão dos atrasos e *jitter* dos pacotes dos vídeos “News 200”, “News 1024”, “Coastguard 200” e “Coastguard 1024” respectivamente, comparando as técnicas *droptail* e ARED (modos byte e pacote).

Fica evidente a capacidade que a técnica ARED possui, especialmente quando operando em modo byte, de manter o atraso médio dos pacotes de vídeo em torno do valor estabelecido como meta (40ms neste caso), bem como de diminuir a variabilidade desse atraso.

O *jitter* foi calculado como a diferença entre o maior e o menor atraso dos pacotes de vídeo ao longo de uma simulação completa. Conforme esperado, percebe-se que a técnica ARED consegue manter os valores de *jitter* mais baixos em relação à técnica *droptail*, especialmente sob condições de congestionamento leve. Porém também nota-se uma tendência desses valores convergirem à medida em que a quantidade de tráfegos concorrentes e o congestionamento aumentam.

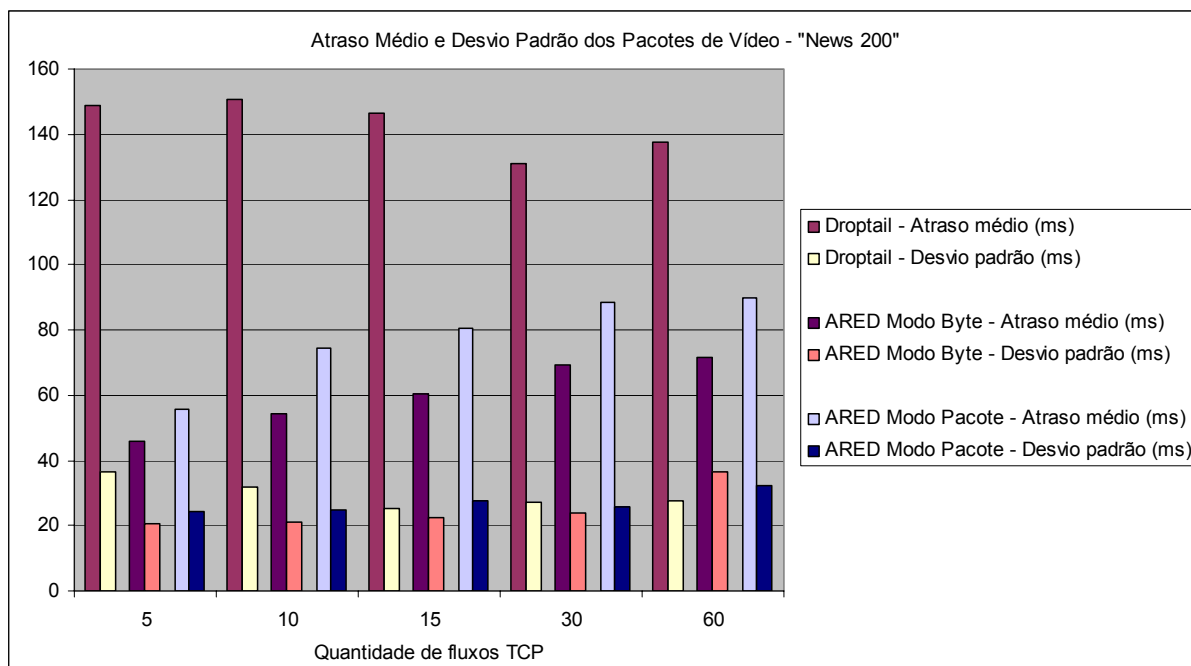


Figura 40 - Atraso médio de rede e desvio padrão dos pacotes do vídeo “News 200”.

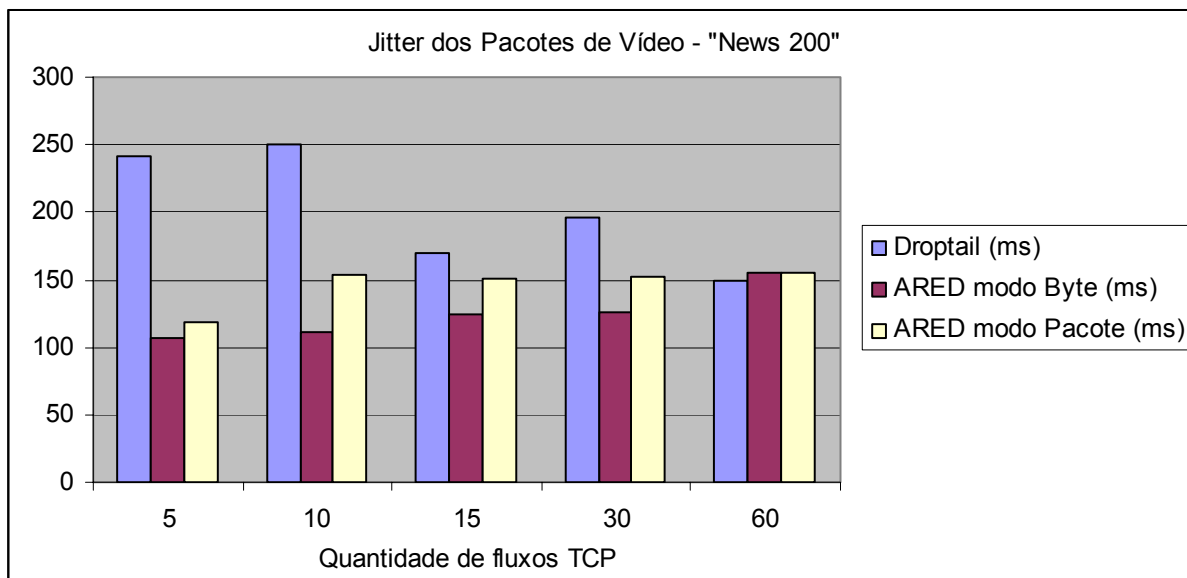


Figura 41 - *Jitter* dos pacotes do vídeo "News 200".

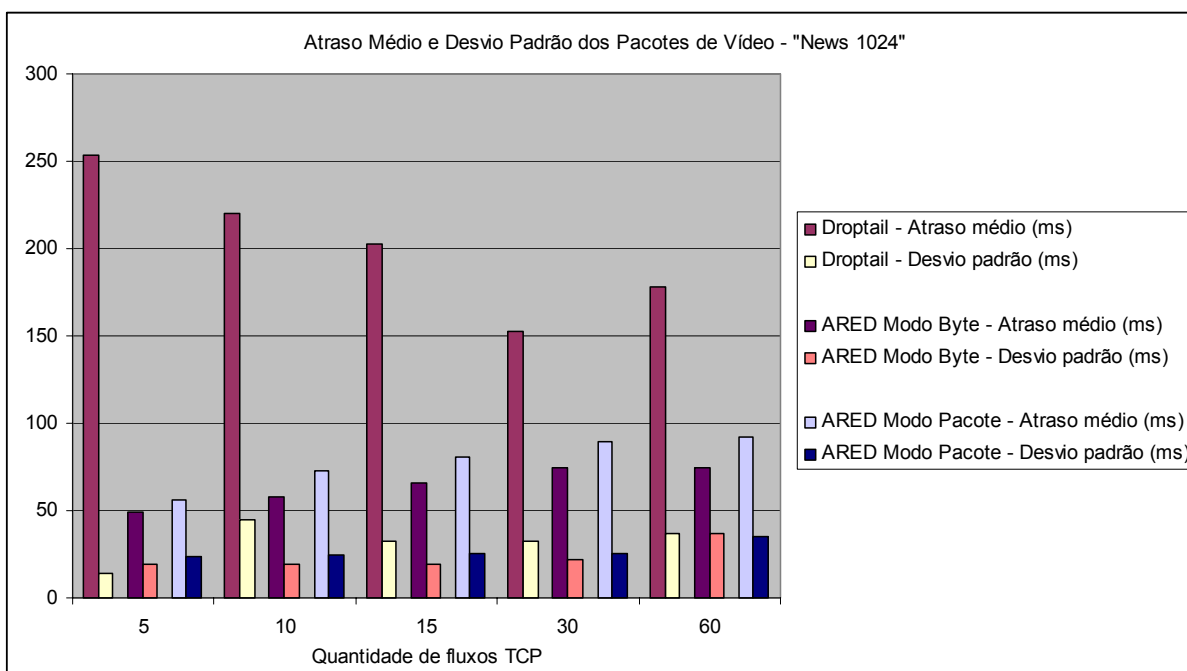


Figura 42 - Atraso médio de rede e desvio padrão dos pacotes do vídeo "News 1024".

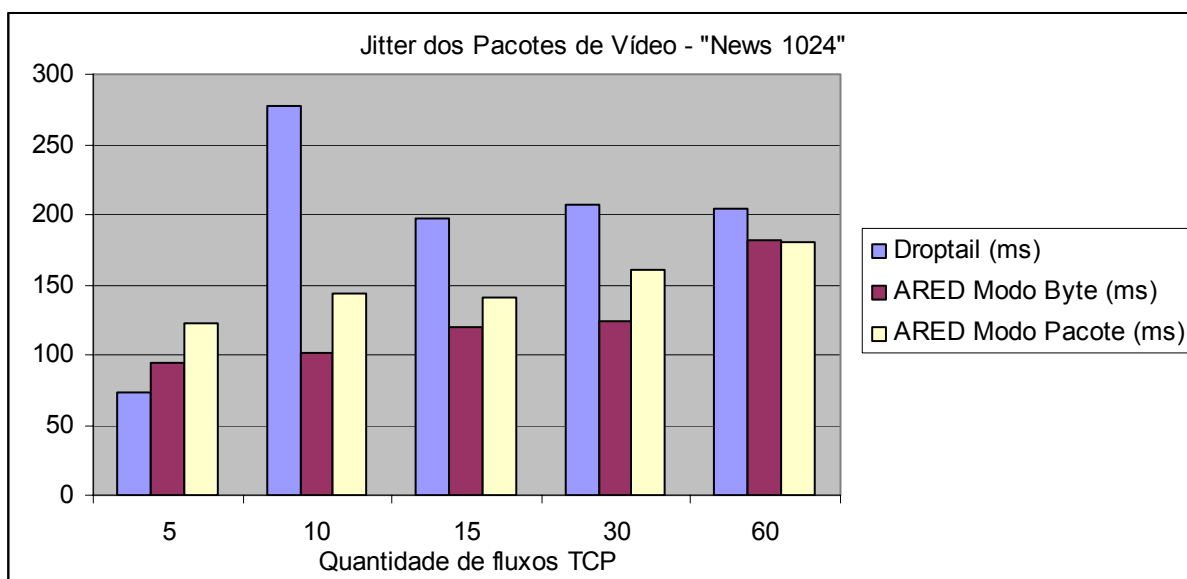


Figura 43 – *Jitter* dos pacotes do vídeo “News 1024”.

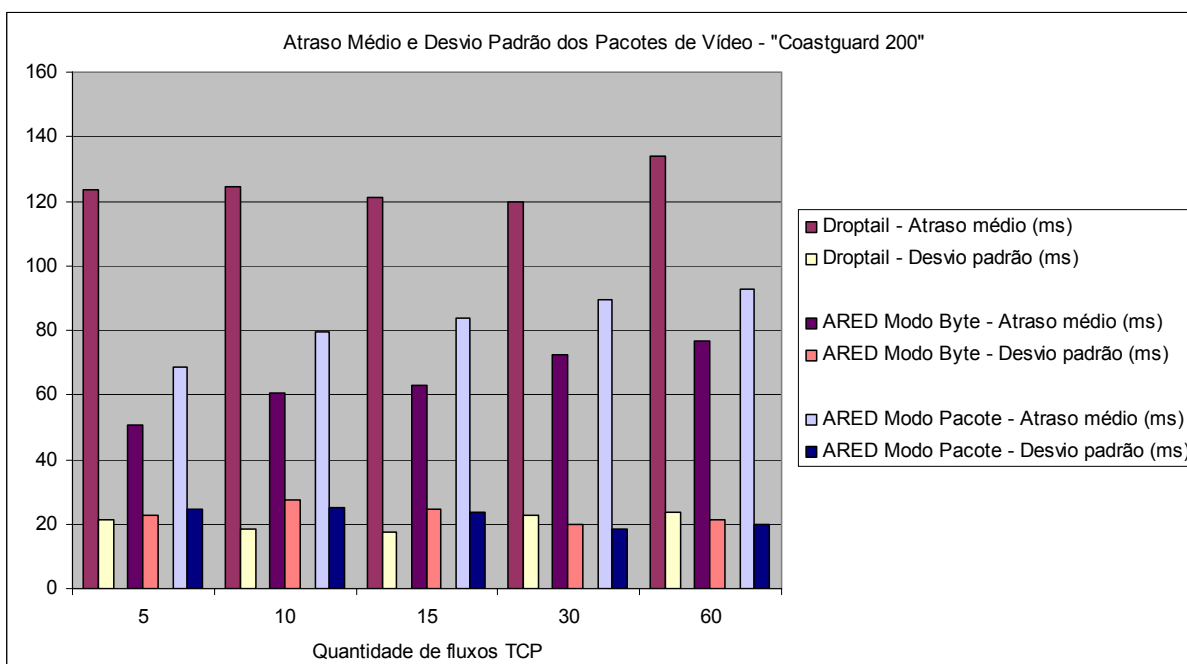


Figura 44 - Atraso médio de rede e desvio padrão dos pacotes do vídeo “Coastguard 200”.

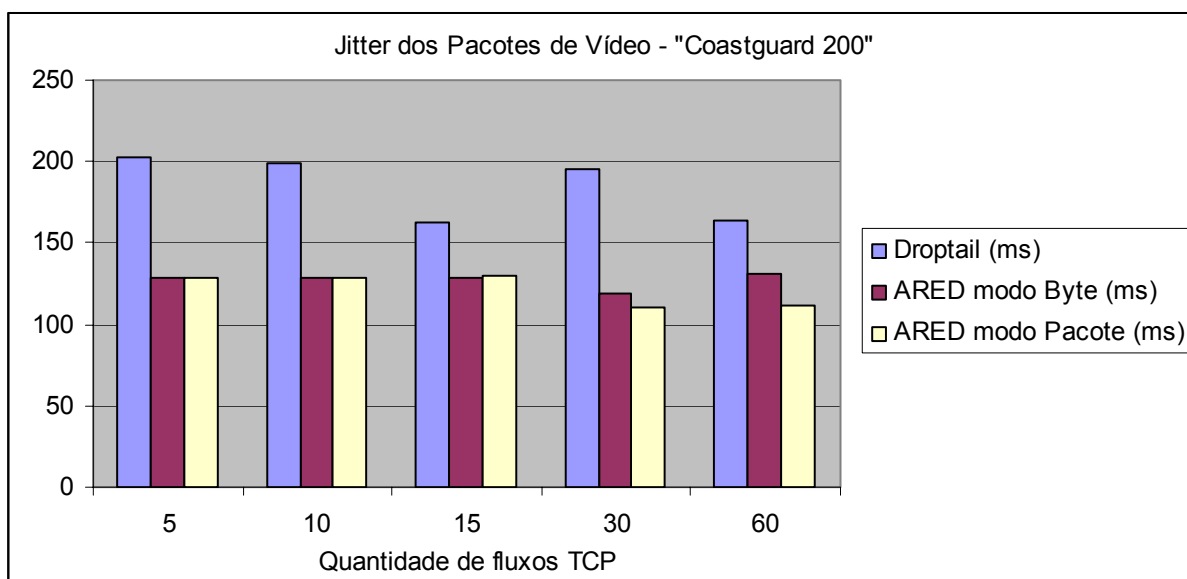


Figura 45 – *Jitter* dos pacotes do vídeo "Coastguard 200".

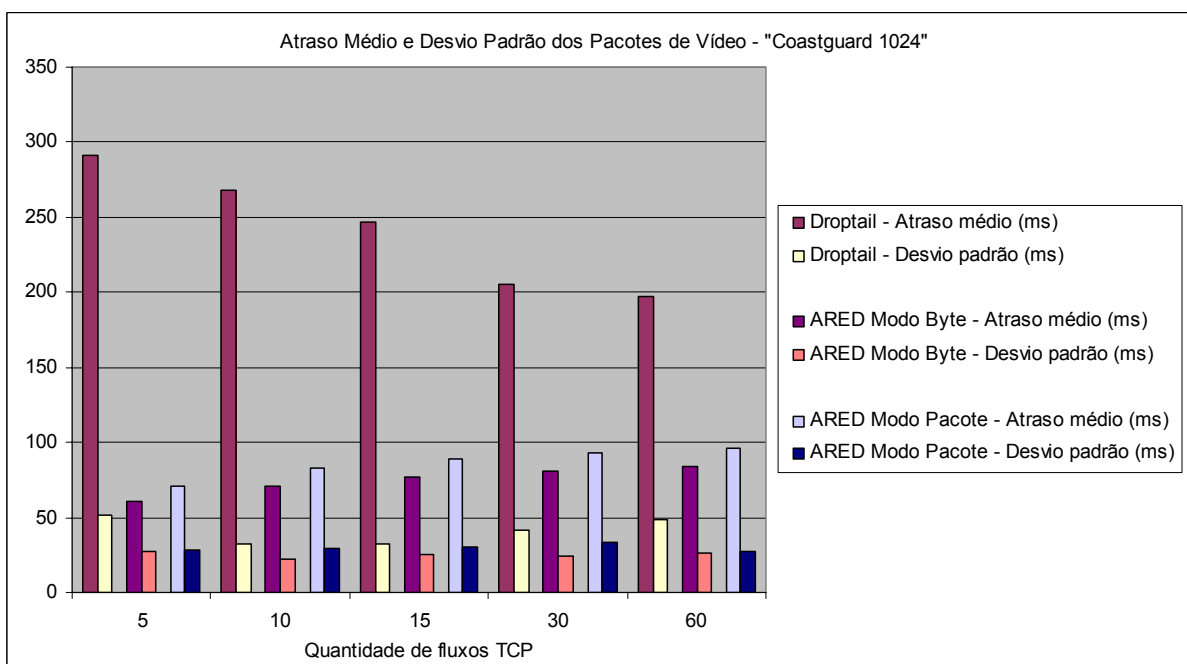


Figura 46 - Atraso médio de rede e desvio padrão dos pacotes do vídeo "Coastguard 1024".

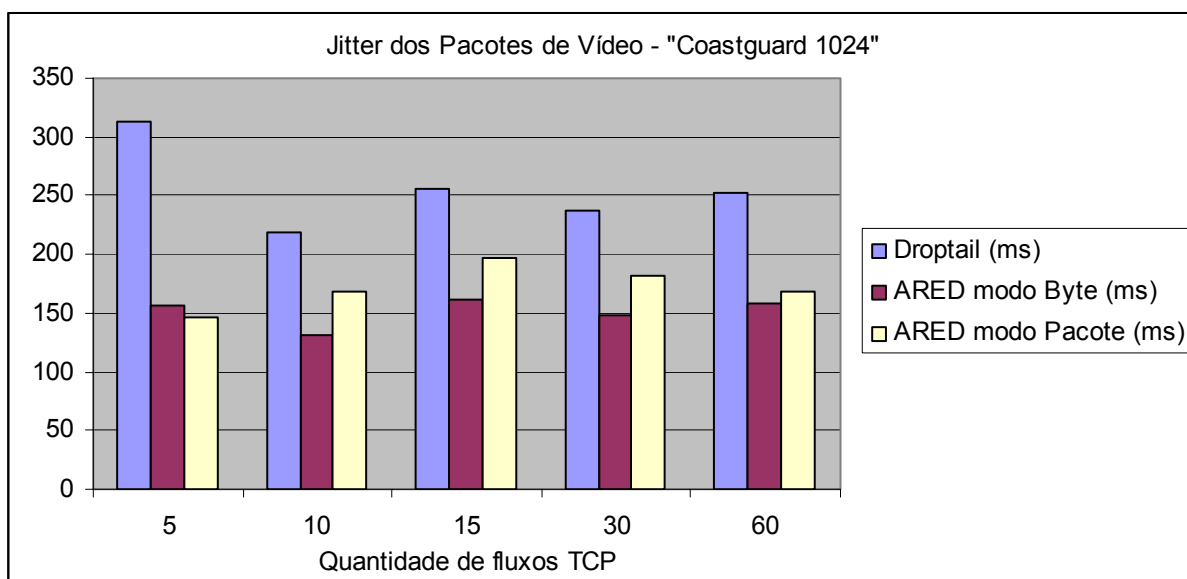


Figura 47 – *Jitter* dos pacotes do vídeo “Coastguard 1024”.

4.4 **Goodput dos Fluxos TCP Concorrentes**

O *goodput* mostra a taxa média de bits atingida pelos fluxos TCP ao final de uma rodada de simulação: somatório (em bits) dos pacotes TCP que chegam ao destino final dividido pelo tempo da simulação.

As figuras 48 e 49 mostram o *goodput* dos fluxos TCP sob as técnicas *droptail* e ARED nos modos byte e pacote. Percebe-se que na técnica *droptail* a tendência dos fluxos TCP é aproveitar a banda que sobra do enlace principal de 2Mbps à medida em que os descartes dos pacotes de vídeo aumentam.

Já na técnica ARED os descartes aleatórios distribuídos no tempo, além de danificarem uma maior quantidade de quadros de vídeo, como visto anteriormente, também afetam negativamente os fluxos TCP ao impedirem que estes ocupem a banda de 2Mbps do enlace principal, apesar das grandes quantidades de descartes provocados nos fluxos de vídeo.

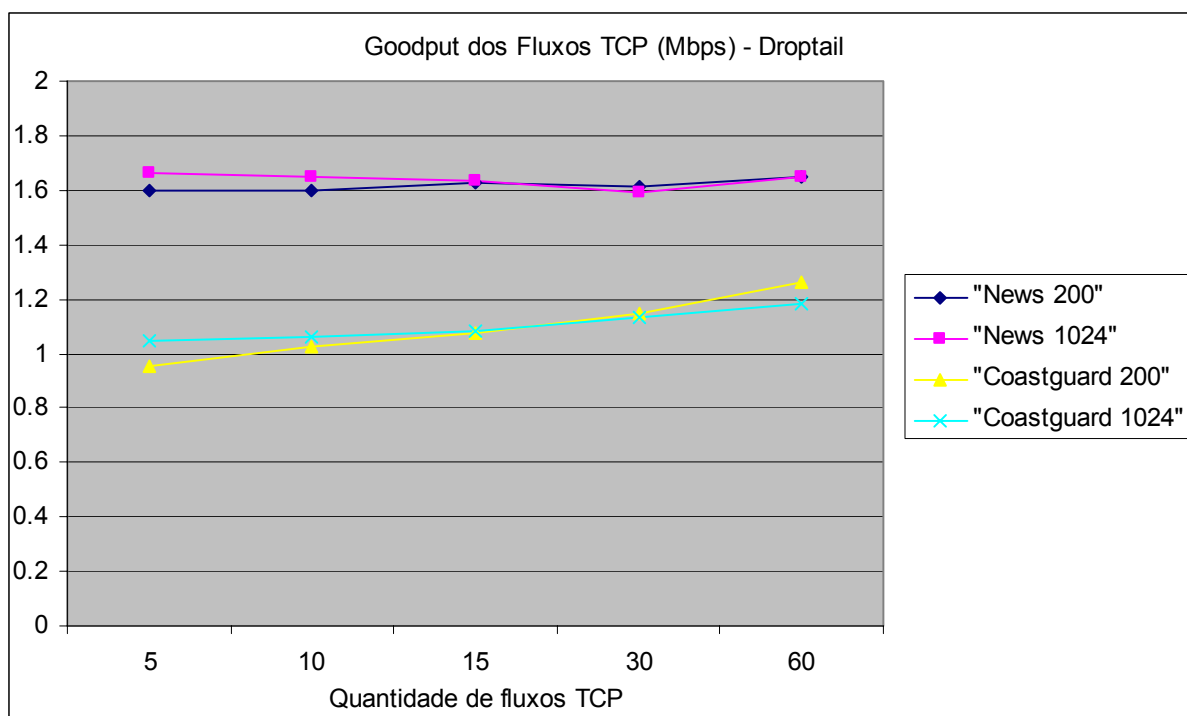


Figura 48 - Goodput dos fluxos TCP sob a técnica *droptail*.

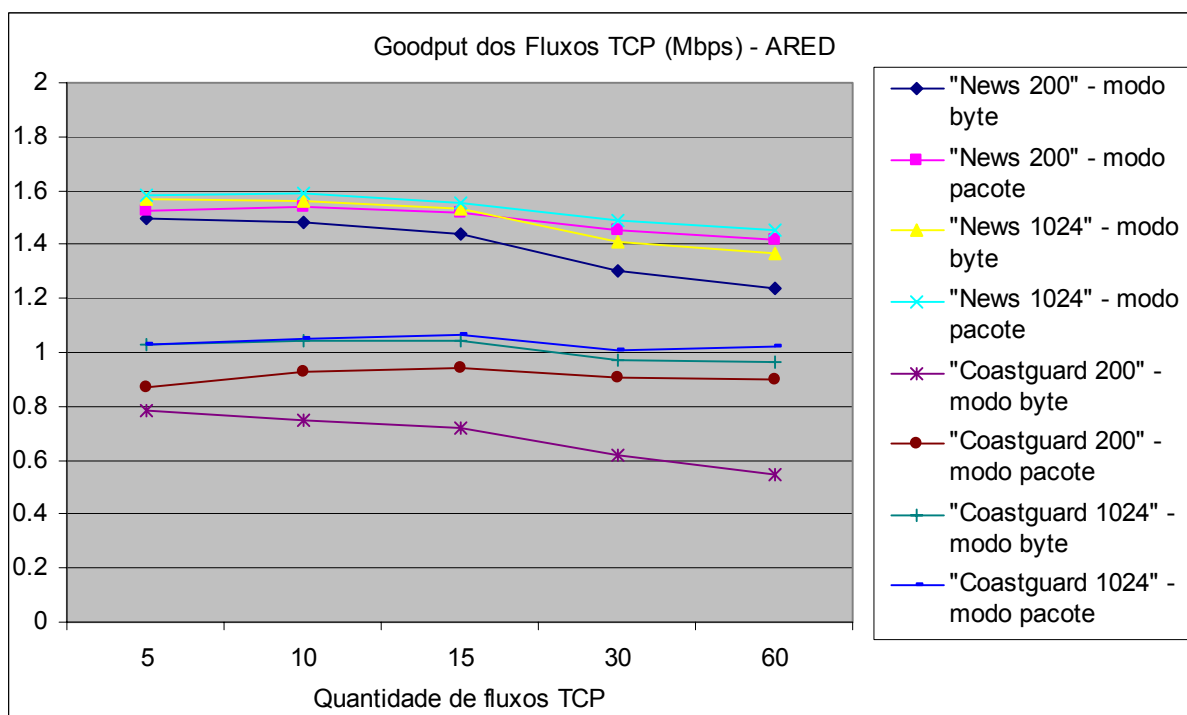


Figura 49 - Goodput dos fluxos TCP sob a técnica ARED.

4.5 PSNR (*Peak Signal-to-Noise Ratio*)

Abaixo são mostradas as medidas de PSNR das sequências de vídeo “News 200”, “News 1024” e “Coastguard 1024”, dentro de cenários que caracterizam bem alguns dos resultados que levaram às conclusões deste trabalho, e nos quais as quantidades de quadros danificados não foram tão altas a ponto de impedir os cálculos de PSNR com a ferramenta que foi utilizada, como aconteceu no caso do vídeo “Coastguard 200”.

4.5.1 Vídeo “News 200”

No caso do vídeo “News 200” dentro do cenário de 10 fluxos TCP concorrentes, onde o desempenho da técnica ARED modo byte foi superior à técnica *droptail* tanto no quesito perdas de pacotes quanto no quesito quadros danificados, percebe-se que as medidas de PSNR de uma sequência de 300 quadros com a aplicação da técnica *droptail* (Figura 50) são superiores às medidas de PSNR com a aplicação da técnica ARED modo byte (Figura 51): média de 13,55 dB para a primeira e 12,65 dB para a segunda.

Tal resultado mostra uma coerência entre as medidas de PSNR e as porcentagens de quadros do tipo I danificados para as duas técnicas mencionadas (vide Figura 36), enquanto que as medidas de pacotes perdidos e quadros danificados indicam desempenho melhor da técnica ARED.

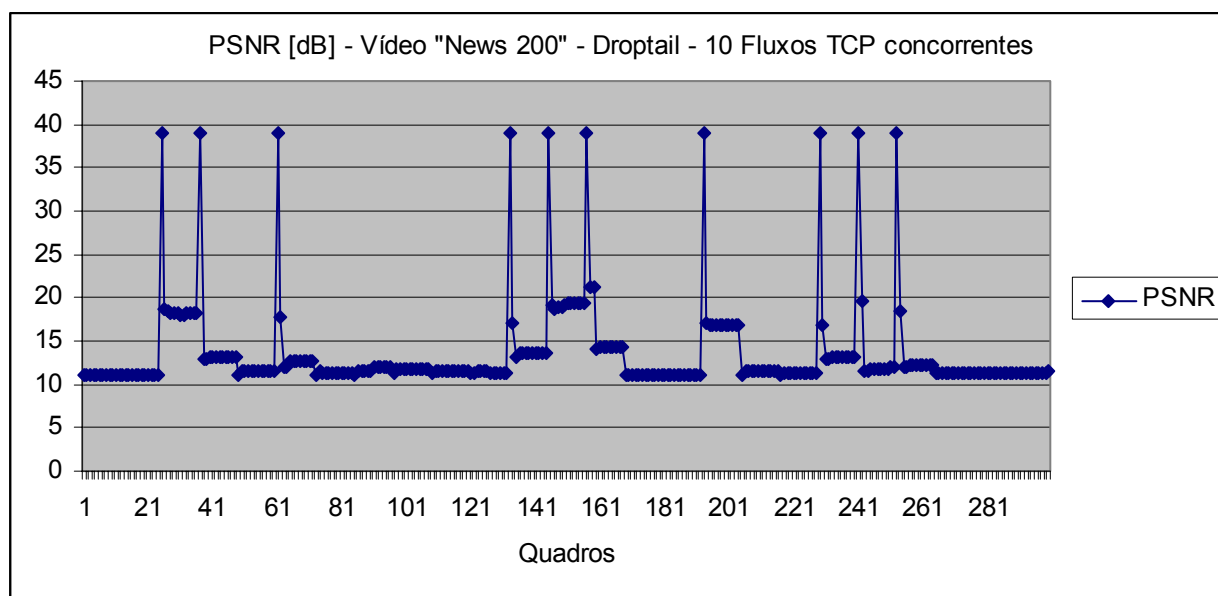


Figura 50 - PSNR de 300 quadros da sequência “News 200” aplicando-se a técnica *droptail* com 10 fluxos TCP concorrentes.

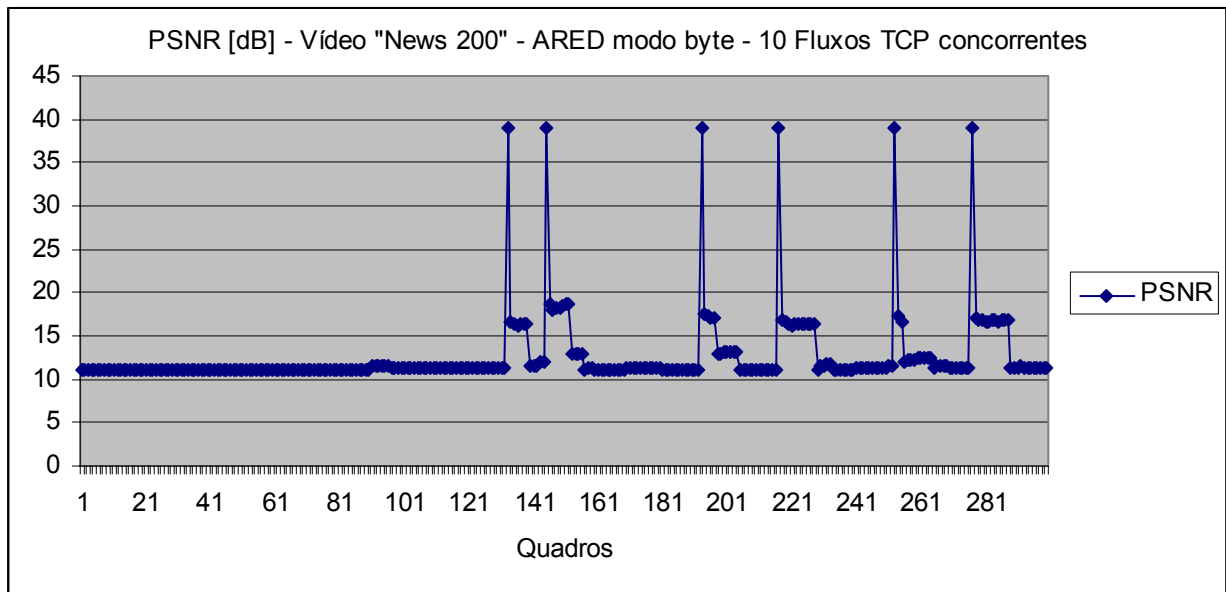


Figura 51 - PSNR de 300 quadros da sequência “News 200” aplicando-se a técnica ARED modo byte com 10 fluxos TCP concorrentes.

4.5.2 Vídeo “News 1024”

Os resultados obtidos para o vídeo “News 1024” dentro do cenário de 15 fluxos TCP concorrentes reforçam a coerência entre as porcentagens de quadros danificados – especialmente os do tipo I - e as medidas de PSNR. A Figura 52 mostra as medidas de PSNR de 300 quadros do vídeo “News 1024” com a aplicação da técnica *droptail*, cuja média de PSNR é 20,75 dB.

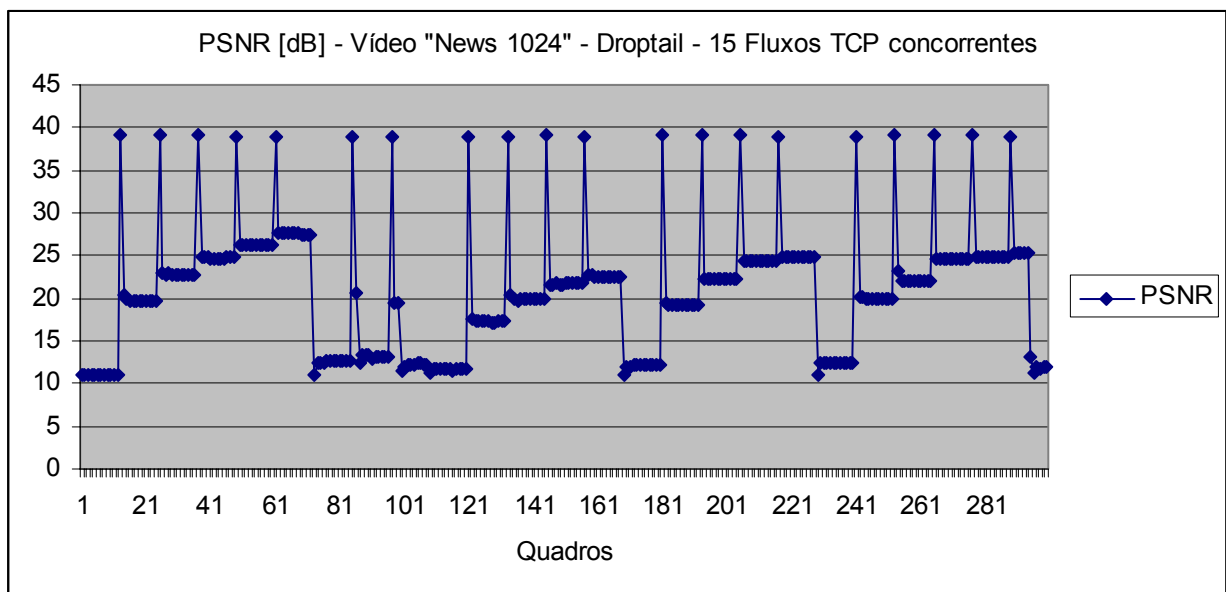


Figura 52 - PSNR de 300 quadros da sequência “News 1024” aplicando-se a técnica *droptail* com 15 fluxos TCP concorrentes.

A Figura 53 mostra as medidas de PSNR dentro do mesmo cenário, porém com a aplicação da técnica ARED modo pacote, cuja média de PSNR é 16,66 dB.

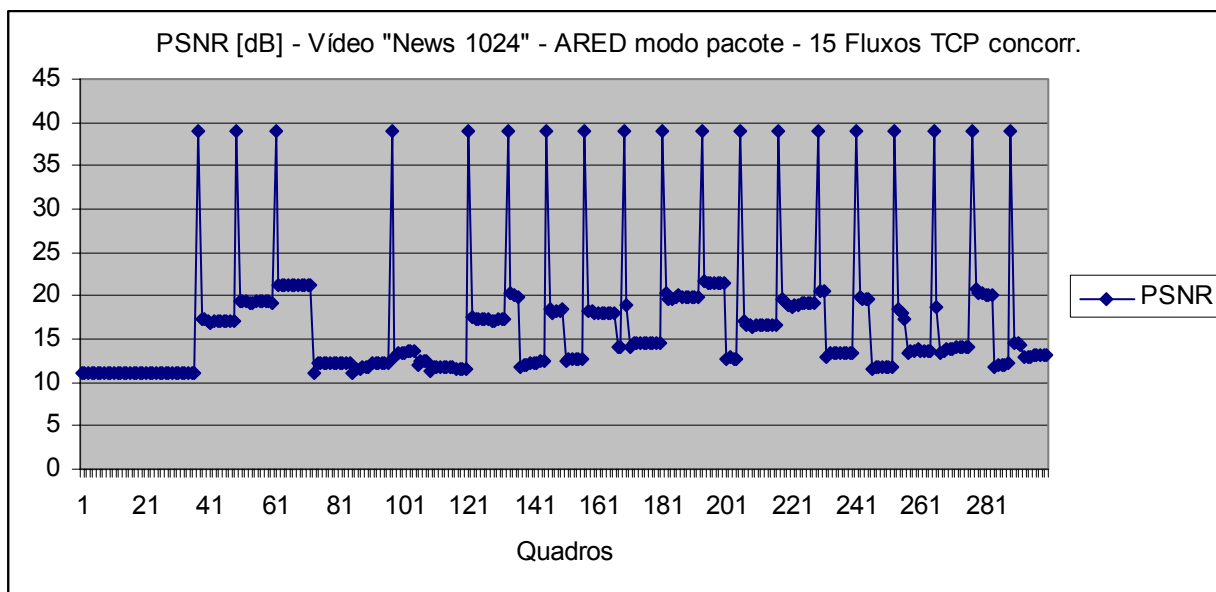


Figura 53 - PSNR de 300 quadros da sequência "News 1024" aplicando-se a técnica ARED modo byte com 15 fluxos TCP concorrentes.

A Figura 54 mostra as medidas de PSNR dentro do mesmo cenário, mas com a aplicação da técnica ARED no modo byte, cuja média de PSNR é 13,30 dB.

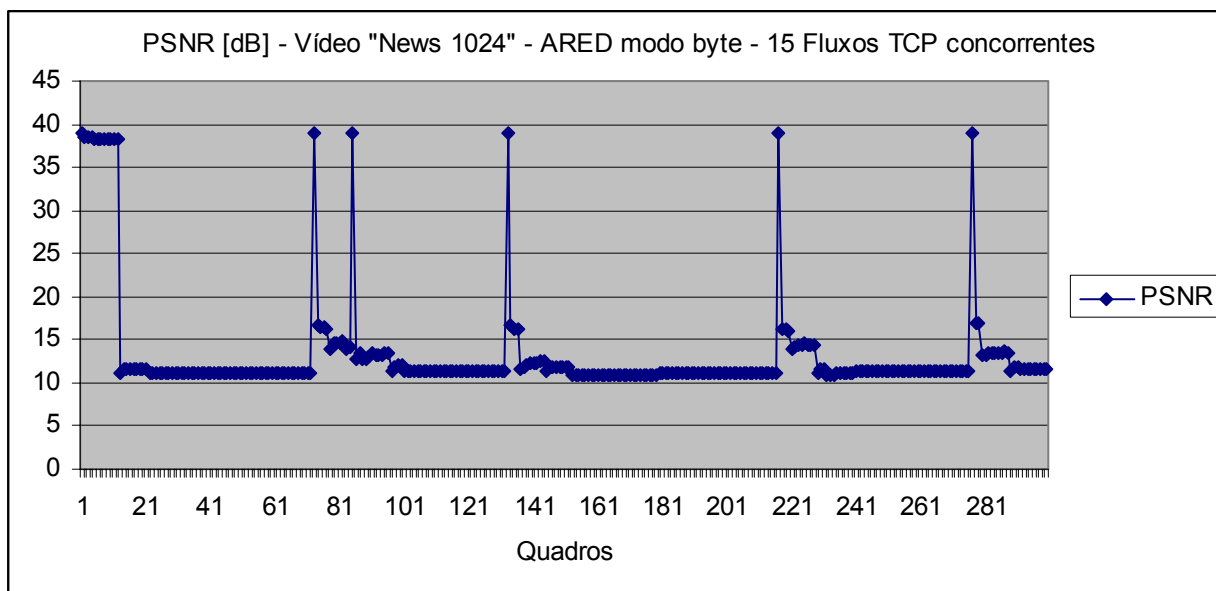


Figura 54 - PSNR de 300 quadros da sequência "News 1024" aplicando-se a técnica ARED modo pacote com 15 fluxos TCP concorrentes.

4.5.3 Vídeo “Coastguard 1024”

Já com respeito ao vídeo “Coastguard 1024”, onde a técnica *droptail* superou a técnica ARED em todos os quesitos, novamente fica perceptível a coerência entre as medidas de PSNR e as porcentagens de quadros do tipo I danificados: média de 19,89 dB com a técnica *droptail* (Figura 55) e 15,26 dB com a técnica ARED modo pacote (Figura 56).

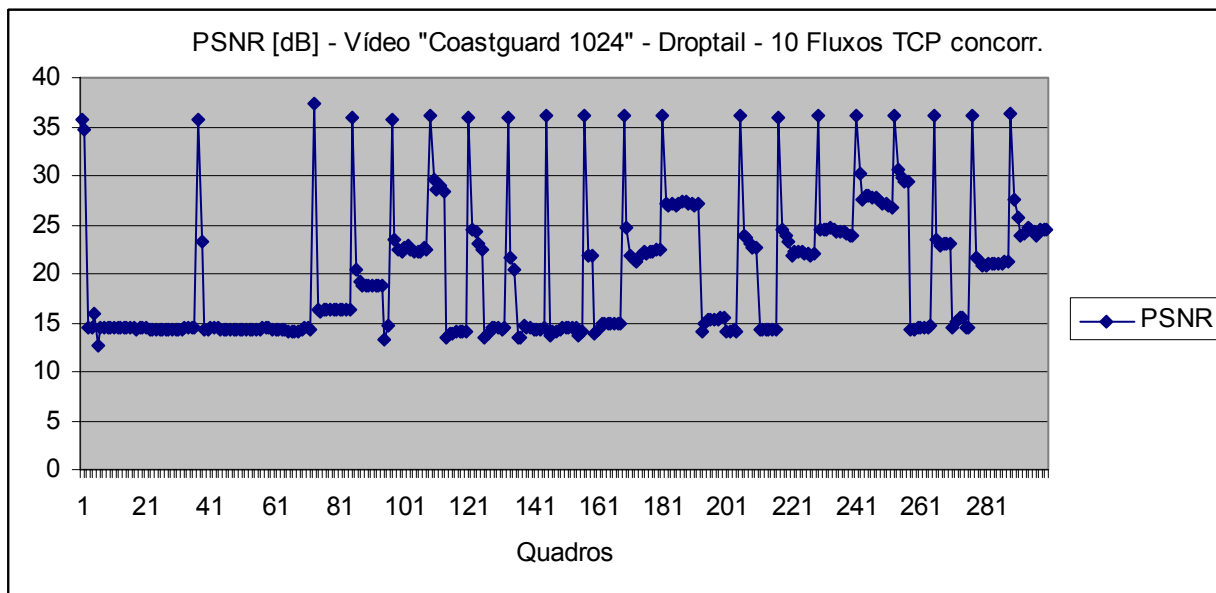


Figura 55 - PSNR de 300 quadros da sequência “Coastguard 1024” aplicando-se a técnica *droptail* com 10 fluxos TCP concorrentes.

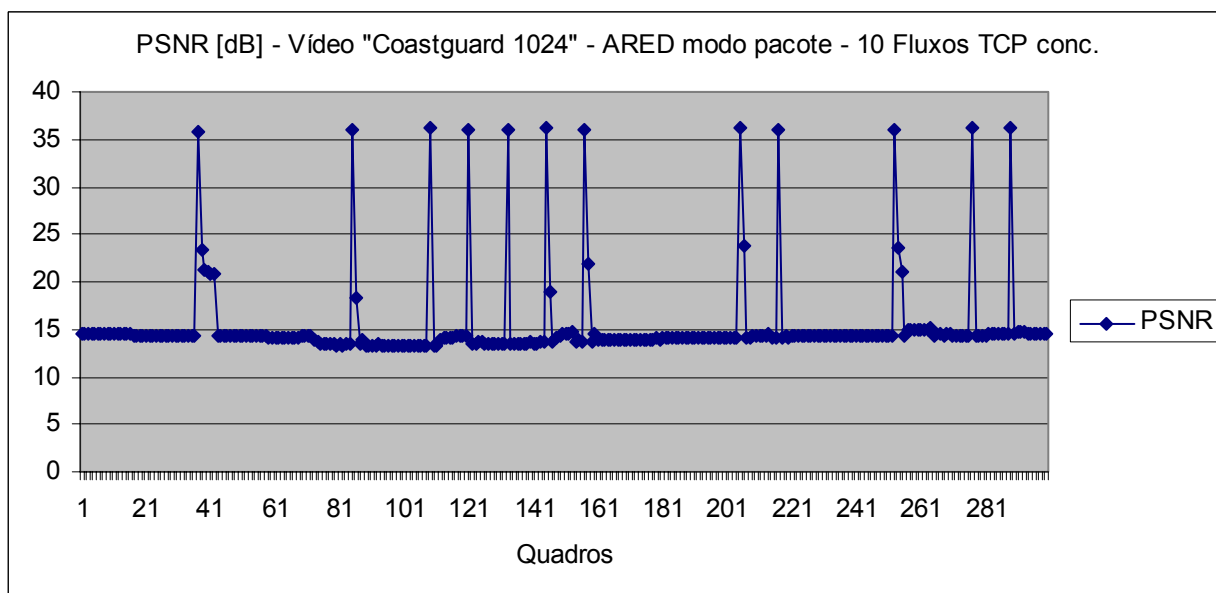


Figura 56 - PSNR de 300 quadros da sequência “Coastguard 1024” aplicando-se a técnica ARED modo pacote com 10 fluxos TCP concorrentes.

5. DISCUSSÃO DOS RESULTADOS

Alguns resultados já esperados foram confirmados e outros, bastante interessantes, foram obtidos a partir das simulações feitas ao longo deste trabalho. Abaixo são revistos e comentados os resultados mais relevantes, seguidos de uma breve discussão.

5.1 Revisão dos Resultados

a) Técnica Droptail

i) Pacotes descartados e quadros danificados

- Para uma mesma sequência de vídeo segmentada em pacotes de diferentes tamanhos, quanto menor o tamanho do pacote, maior a porcentagem de pacotes descartados. Isso se explica pela maior quantidade de pacotes pequenos gerados pela fonte de vídeo, os quais fazem com que a fila, medida em pacotes, seja mais disputada.
- Para uma mesma quantidade de fluxos TCP concorrentes, quanto menor o tamanho do pacote de vídeo, maior a porcentagem de quadros danificados, pois as informações referentes aos quadros estão distribuídas em um número maior de pacotes.
- As inclinações das curvas de pacotes descartados e quadros danificados são muito parecidas, independentemente do tamanho dos segmentos dos quadros de vídeo. Isso pode ser explicado pelas flutuações no final da fila, que provocam perdas em rajadas, danificando um único quadro ou uma sequência de quadros adjacentes.

ii) Atrasos médios e *jitter*

- Quanto aos atrasos médios dos pacotes de vídeo, os mesmos permanecem praticamente inalterados no cenário de pacotes pequenos (200 bytes), enquanto demonstram uma tendência de queda no cenário de pacotes grandes (1024 bytes) à medida em que o número de fluxos TCP concorrentes aumenta. Tais resultados podem ser explicados em parte pelo maior número de pacotes de vídeo descartados: no caso dos pacotes grandes, muitos descartes fazem com que o tamanho médio dos pacotes diminua, diminuindo também o tempo de enfileiramento; no caso dos pacotes pequenos, os descartes não provocam alteração importante no tamanho médio dos pacotes.

- Com respeito ao jitter, houve uma leve tendência de diminuir os valores à medida em que o nível de congestionamento aumenta, pois as flutuações no final da fila tornam-se menos acentuadas.

iii) *Goodput* dos fluxos TCP concorrentes

- Percebe-se que os fluxos TCP concorrentes conseguem aproveitar praticamente toda a banda que não é usada pelo fluxo de vídeo. A explicação para isso é que as perdas em rajadas do *droptail* provocam a redução da taxa de um número limitado de fluxos TCP, enquanto os demais continuam enviando pacotes a taxas crescentes, até que um eventual descarte venha a afetá-los.

b) Técnica ARED

i) Pacotes descartados e quadros danificados

- No modo byte, a técnica ARED favorece significativamente os pacotes de tamanho pequeno, o que está de acordo com o comportamento esperado da mesma.
- No entanto, no modo pacote não há um favorecimento dos pacotes grandes, como se poderia esperar. Na verdade, os pacotes grandes sofrem descartes a taxas muito parecidas, seja no modo byte ou no modo pacote.
- As inclinações das curvas de pacotes descartados e quadros danificados são diferentes, sendo que a curva de quadros danificados apresenta uma tendência de subida mais pronunciada. Isso pode ser explicado pelos descartes distribuídos no tempo que a técnica ARED provoca, afetando um maior número de quadros de vídeo.

ii) Atrasos médios e jitter

- Em todos os cenários houve uma leve tendência ao aumento dos atrasos médios dos pacotes de vídeo à medida em que a quantidade de fluxos TCP concorrentes aumenta. Também é perceptível um melhor desempenho da técnica operando em modo byte, independentemente do tamanho dos pacotes de vídeo.
- Quanto ao jitter, praticamente não houve alterações significativas nos diferentes cenários, o que demonstra a grande capacidade que a técnica ARED possui de manter tal parâmetro estável.

iii) *Goodput* dos fluxos TCP concorrentes

- Os fluxos TCP concorrentes não conseguem aproveitar toda a banda que não é usada pelo fluxo de vídeo, pois uma proporção maior dos fluxos TCP é afetada pelos descartes aleatórios distribuídos no tempo da técnica ARED e forçados a reduzir suas taxas de envio de pacotes.

c) Comparativo entre *droptail* e ARED

- No quesito descarte de pacotes, quando os pacotes de vídeo são pequenos, a técnica ARED, especialmente no modo byte, tem melhor desempenho, conforme esperado. Já quando os pacotes de vídeo são grandes, os desempenhos são muito parecidos, sendo que a técnica *droptail* apresenta ligeira vantagem.
- No quesito quadros danificados, percebe-se um desempenho melhor da técnica *droptail* em todos os cenários, exceto um: vídeo “News 200”, com a técnica ARED operando em modo byte. Esse resultado aparece graças a um conjunto de características favoráveis à aplicação da técnica ARED: pacotes de tamanho pequeno e (relativamente) baixa taxa de bits, proporcionalmente à capacidade do enlace gargalo simulado.
- No quesito quadros do tipo I danificados, a técnica *droptail* obteve um desempenho melhor em todos os cenários, especialmente naquele em que os pacotes de vídeo são grandes (“News 1024” e “Coastguard 1024”). Considerando-se a importância dos quadros tipo I para a decodificação das sequências de vídeo, trata-se de um resultado bastante relevante.
- Numa comparação direta, uma mesma porcentagem de pacotes descartados danifica uma porcentagem maior de quadros quando a técnica usada é ARED, o que se explica pela distribuição dos descartes em intervalos regulares de tempo, ao contrário dos descartes em rajadas do *droptail*.
- Quanto aos atrasos médios e variações de atraso, a técnica ARED claramente tem um melhor desempenho que a técnica *droptail*, mantendo os atrasos em níveis mais baixos e com menos variações.
- Quanto ao *goodput* dos fluxos TCP concorrentes, percebe-se que a técnica *droptail* permite que a banda não utilizada pelo fluxo de vídeo seja melhor aproveitada.
- Quanto às medidas de PSNR, fica clara a proximidade da relação entre quadros do tipo I danificados e PSNR. Em outras palavras, o parâmetro que mais se aproxima de uma medida objetiva da qualidade dos vídeos transmitidos não é a

perda de pacotes nem a quantidade de quadros danificados de uma foram geral, mas sim a quantidade de quadros do tipo I danificados.

5.2 Discussão dos Resultados

Apesar de a técnica ARED ter sido projetada para operar em conjunto com fluxos responsivos, num cenário de mistura de fluxos responsivos e irresponsivos, os fluxos irresponsivos (como os fluxos de VoIP e vídeo discutidos nas seções anteriores) podem se beneficiar das menores flutuações das filas e menores porcentagens de pacotes descartados.

No entanto, o quanto os fluxos irresponsivos se beneficiam com a técnica ARED varia de acordo com as características dos mesmos, especialmente no que diz respeito à parcela do enlace gargalo ocupada por tais fluxos, ao tamanho dos pacotes que os compõem e à forma como as informações estão distribuídas nos pacotes. Como demonstrado por Vitalio *et al.* [4], os fluxos VoIP têm um ganho qualitativo bastante importante com a aplicação da técnica ARED em lugar da *droptail*, mas o mesmo não se pode dizer dos fluxos de vídeo simulados neste trabalho.

Comparativamente, os fluxos VoIP geram baixas taxas de bits e são compostos de pacotes pequenos e independentes, onde cada pacote transporta alguns milissegundos de voz. Já os fluxos de vídeo codificados no padrão H.264/MPEG-4 AVC aqui simulados geram altas taxas de bits e um grande número de pacotes interdependentes, de acordo com a estrutura dos GOPs definida no momento da codificação.

O que se percebe ao longo das simulações é que os fluxos de vídeo são prejudicados pela aplicação da técnica ARED, especialmente se considerarmos a constatação de que uma mesma porcentagem de pacotes descartados danifica mais quadros de vídeo quando a técnica ARED é aplicada. Também considerando a maior importância que os quadros do tipo I possuem em relação aos quadros do tipo P, um maior número de quadros tipo I danificados com a aplicação da técnica ARED resulta em uma medida de PSNR mais baixa.

Obviamente, é importante validar tais percepções com um mapeamento das medições objetivas em escalas subjetivas, procurando medir a avaliação que um usuário real faria da transmissão de vídeo simulada.

6. CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho de pesquisa com fluxos de vídeo H.264/MPEG-4 AVC transmitidos sobre redes IP *best effort* teve como motivação os expressivos ganhos qualitativos obtidos para fluxos VoIP demonstrados por Vitalio *et al.* [4] a partir da aplicação da técnica ARED no lugar da técnica *droptail* em enlaces do tipo gargalo simulados no NS2.

Utilizando uma topologia de rede semelhante, foi avaliado o impacto na qualidade de serviço dos referidos fluxos de vídeo em diferentes cenários, medindo as porcentagens de pacotes descartados, quadros danificados, PSNR, atrasos médios e *jitter* quando os mesmos disputam um enlace do tipo gargalo com fluxos TCP.

Foi verificado que, se por um lado as porcentagens de pacotes de vídeo descartados realmente são menores com técnica ARED nos cenários de pacotes pequenos, por outro uma mesma quantidade de pacotes descartados afeta mais quadros de vídeo com essa técnica do que com a técnica *droptail* em todos os cenários simulados.

Além disso, os fluxos de vídeo H.264/MPEG-4 AVC geram quadros com diferentes graus de importância para a decodificação. Foi verificado que os quadros do tipo I (os mais importantes) são danificados num número expressivamente maior com a aplicação da técnica ARED.

No que diz respeito a atrasos e variações de atraso, devido a não existirem restrições de temporização tão fortes para fluxos de vídeo não-conversacionais como existem para os fluxos VoIP, o ganho qualitativo obtido pela técnica ARED nas condições simuladas é de pequena importância.

Quanto à ocupação do enlace, verificamos que, dentro das condições simuladas, a técnica ARED impediu que os fluxos TCP concorrentes ocupassem a banda não ocupada pelo fluxo de vídeo, enquanto que tal ocupação ocorreu com a técnica *droptail*.

Com respeito à medida de PSNR média das sequências de vídeo, foi percebido que a porcentagem de quadros do tipo I danificados é o parâmetro que apresenta uma relação mais próxima com a referida medida. Tal resultado demonstra que a porcentagem de pacotes perdidos e quadros (independentemente do tipo) danificados não são indicativos tão bons da qualidade do vídeo após a transmissão quanto a porcentagem de quadros do tipo I danificados.

Considerando o exposto acima, é questionável a aplicação da técnica ARED com o objetivo de melhorar a qualidade de fluxos de vídeo no formato H.264/MPEG-4 AVC trafegando por enlaces congestionados. Diferentemente do que foi demonstrado por Vitalio *et al.* [4] em relação aos fluxos VoIP, numa situação de grande congestionamento na rede a técnica ARED degrada

mais a qualidade dos fluxos de vídeo, comparada à técnica *droptail*, como resultado de sua característica de distribuir os descartes de pacotes ao longo do tempo.

Obviamente a real percepção de um usuário no que diz respeito à qualidade do vídeo transmitido dentro das condições simuladas neste trabalho exige um mapeamento das medições objetivas de perdas de pacotes de vídeo, atrasos e *jitter* numa escala de avaliação subjetiva. Por isso, fica aqui a sugestão de realizar tal avaliação como trabalho futuro.

Também como trabalho futuro sugere-se a avaliação do esforço computacional necessário para recuperar as informações de vídeo perdidas de forma a melhorar a qualidade percebida. O que é mais viável: recuperar mais informação de vídeo de uma quantidade menor de quadros danificados (técnica *droptail*) ou recuperar menos informação de vídeo de uma quantidade maior de quadros danificados (técnica ARED)?

No entanto é importante enfatizar que a complexidade de redes IP reais, sejam cabeadas ou sem-fio, não pode ser captada integralmente a partir de simulações computacionais. Dessa forma, sugerimos o aprofundamento da pesquisa sobre esta e outras técnicas de AQM em redes IP reais, preferencialmente com implementações reais dos respectivos algoritmos.

7. BIBLIOGRAFIA

- [1] S. Kwon, A. Tamhankar, K. Rao, "Overview of H.264/MPEG-4 Part 10", J. Visual Communication and Image Representation, vol. 17, 2006.
- [2] M. N. Zapater, G. Bressan, "A Proposed Approach for Quality of Experience Assurance for IPTV", IEEE Proceedings of the First International Conference on the Digital Society, 2007.
- [3] White paper - Shenick Network Systems, "Testing MPEG based IP video QoE/QoS", disponível em http://www.shenick.com/media_lib/files/Testing_MPEG_based_IP_video_by_Shenick.pdf.
- [4] Reguera, Vitalio A., Paliza, Félix F. A., Fernández, Evelio M. G., and Godoy Jr., Walter, "Voice over IP Quality of Service Using Active Queue Management" Proceedings of the VI International Telecommunications Symposium (ITS2006), Fortaleza-CE, Brasil, September de 2006.
- [5] S. McCanne and S. Floyd. ns Network Simulator. <http://www.isi.edu/nsnam/ns/>.
- [6] A. Tanenbaum, "Computer Networks – 4th Edition", Prentice Hall, 2003.
- [7] W. Stallings, "Data and Computer Communications", 2003.
- [8] S. Wenger, "H.264/AVC Over IP", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, 2003.
- [9] M. N. dos Santos, "Medidas de qualidade de voz em redes IP", dissertação de Mestrado, UFPR, 2006.
- [10] H. Rui, C. Li, S. Qiu, "Evaluation of Packet Loss Impairment on Streaming Vídeo", Journal of Zhejiang University Science B, 2006.
- [11] Reguera, Vitalio A., Paliza, Félix F. A., Godoy Jr., Walter, and Fernández, Evelio M. G. "On the Impact of Active Queue Management on Voice over IP Quality of Service", Computer Communications, Vol. 31, No.1, pp. 73-87, January 2008.
- [12] S. Qiu, H. Rui, L. Zhang, "No-reference Perceptual Quality Assessment for Streaming Video Based on Simple End-to-end Network Measures", ICNS'06, IEEE, 2006.
- [13] G. Davini, D. Quaglia, J.C. De Martin, C. Casetti, "Perceptually-Evaluated Loss-Delay Controlled Adaptive Transmission of MPEG Vídeo over IP", IEEE, 2003.
- [14] K. Kerpez, D. Waring, G. Lapiotis, J. B. Lyles, R. Vaidyanathan, "IPTV Service Assurance", IEEE Communications Magazine, 2006.
- [15] G. Sullivan, T. Wiegand, "Video Compression – From Concepts to the H.264/AVC Standard", Proceedings of the IEEE, Vol. 93, No. 1, 2005.
- [16] White paper - The MPEG Industry Forum, "Understanding MPEG-4: Technologies, Advantages, and Markets", 2005.
- [17] White paper - Envivio, "IP Streaming of MPEG-4: Native RTP vs MPEG-2 Transport Stream", 2005.
- [18] G. Auwera, P. David, M. Reisslein, "Traffic Characteristics of H.264/AVC Variable Bit Rate Video", accepted in IEEE Communications Magazine, 2008.
- [19] G. Auwera, P. David, M. Reisslein, "Traffic and Quality Characterization of Single-Layer Video Streams Encoded with the H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding

Extension”, accepted in IEEE Transactions on Broadcasting: Quality Issues in Multimedia Broadcasting, 2008.

[20] H.264/AVC Software Coordination, Reference Software JM 13.0 [Online], disponível em <http://iphome.hhi.de/suehring/ttml/>.

[21] Chih-Heng Ke, Ce-Kuen Shieh, Wen-Shyang Hwang, Artur Ziviani, “An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission”, Journal of Information Science and Engineering (JISE); Vol: 24; No: 2; Pág: 425-440; 2008.

[22] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance”, IEEE/ACM Trans. Networking, vol. 1, no. 4, August 1993, pp. 397–413.

[23] S. Floyd, R. Gummadi, S. Shenker, “Adaptive RED: An Algorithm for Increasing the Robustness of RED’s Active Queue Management”, Available online: <http://www.icir.org/floyd/papers/adaptiveRed.pdf>, August 1, 2001.

[24] A. Ziviani, J. Rezende, O. Duarte, S. Fdida, “Improving the Delivery Quality of MPEG Video Streams by Using Differentiated Services”, IEEE Potentials, January/February 2007, pp. 16-19.

[25] Video Trace Library, Arizona State University [Online], disponível em <http://trace.eas.asu.edu/yuv/index.html>.

[26] VIDEOHelp.com - Forum, Guides, Tools [Online], disponível em <http://www.videohelp.com/tools/MP4Box>.

[27] J. Klaue, B. Rathke, A. Wolisz, “EvalVid – A Framework for Video Transmission and Quality Evaluation”, 13th International Conference on Modeling Techniques and Tools for Computer Evaluation, Urbana, Illinois, USA, 2003.

Anexo I

Script base das simulações no NS2

```
#INPUT ns simu.tcl [arquivo video] [droptail ou ARED] [fluxos TCP]
#-----
#DEFINIÇÕES
set video [lindex $argv 0]
set aqm [lindex $argv 1]
set flows [lindex $argv 2]
set ns [new Simulator]

set tracefile1 [open out.tr w]
$ns trace-all $tracefile1

$ns color 1 Blue
$ns color 2 Red
#set namfile [open out.nam w]
#$ns namtrace-all $namfile

set max_fragmented_size 1500
set packetSize 1500
#-----
#TOPOLOGIA FIXA
set r1 [$ns node]
set r2 [$ns node]
set vi1 [$ns node]
set vi2 [$ns node]
$ns duplex-link $vi1 $r1 10Mb 2ms DropTail
$ns duplex-link $vi2 $r2 10Mb 2ms DropTail

if {$aqm == 1} {
Queue/RED set adaptive_ true
Queue/RED set targetdelay_ 0.04
Queue/RED set queue_in_bytes false
Queue/RED set mean_pktsize_ 455
Queue/RED set setbit_ true
Queue/RED set bytes_ false
$ns duplex-link $r1 $r2 2Mb 5ms RED
$ns queue-limit $r1 $r2 120
} else {
$ns duplex-link $r1 $r2 2Mb 5ms DropTail
$ns queue-limit $r1 $r2 120
}

if {$aqm == 1} {
set udpl [new Agent/myUDP]
$ns attach-agent $vi1 $udpl
$udpl set packetSize_ $packetSize
$udpl set_filename ($video)_PAC_RED__sd.txt
$udpl set fid_ 1

set null1 [new Agent/myUdpSink2]
$ns attach-agent $vi2 $null1
$ns connect $udpl $null1
$null1 set_trace_filename ($video)_PAC_RED__rd.txt

set trace_file_name $video
set trace_file [new Tracefile]
$trace_file filename $trace_file_name
set video1 [new Application/Traffic/myTrace2]
$video1 attach-agent $udpl
$video1 attach-tracefile $trace_file
} else {
set udpl [new Agent/myUDP]
```



```

$ns attach-agent $vi1 $udpl
$udpl set packetSize_ $packetSize
$udpl set_filename ($video)__dp_sd.txt
$udpl set fid_ 1

set null1 [new Agent/myUdpSink2]
$ns attach-agent $vi2 $null1
$ns connect $udpl $null1
$null1 set_trace_filename ($video)__dp_rd.txt

set trace_file_name $video
set trace_file [new Tracefile]
$trace_file filename $trace_file_name
set video1 [new Application/Traffic/myTrace2]
$video1 attach-agent $udpl
$video1 attach-tracefile $trace_file
}
#-----
#TCPs
#-----
#TPC1

for {set i 0} {$i < $flows} {incr i} {
    set s($i) [$ns node]
    set k($i) [$ns node]
    $ns duplex-link $k($i) $r1 10Mb 10ms DropTail
    $ns duplex-link $s($i) $r2 10Mb 10ms DropTail
    set tcp1($i) [new Agent/TCP]
    $tcp1($i) set packetSize_ 120
    $ns attach-agent $s($i) $tcp1($i)
    set sink1($i) [new Agent/TCPSink]
    $ns attach-agent $k($i) $sink1($i)
    $ns connect $tcp1($i) $sink1($i)
    set ftp1($i) [new Application/FTP]
    $ftp1($i) attach-agent $tcp1($i)
    $ns at 0.1 "$ftp1($i) start"
    $ns at 49.9 "$ftp1($i) stop"
}

#-----
#TPC2
for {set i 0} {$i < $flows} {incr i} {
    set t($i) [$ns node]
    set l($i) [$ns node]
    $ns duplex-link $t($i) $r1 10Mb 17ms DropTail
    $ns duplex-link $l($i) $r2 10Mb 17ms DropTail
    set tcp2($i) [new Agent/TCP]
    $tcp2($i) set packetSize_ 300
    $ns attach-agent $t($i) $tcp2($i)
    set sink2($i) [new Agent/TCPSink]
    $ns attach-agent $l($i) $sink2($i)
    $ns connect $tcp2($i) $sink2($i)
    set ftp2($i) [new Application/FTP]
    $ftp2($i) attach-agent $tcp2($i)
    $ns at 0.3 "$ftp2($i) start"
    $ns at 49.9 "$ftp2($i) stop"
}

#-----
#TPC3
for {set i 0} {$i < $flows} {incr i} {
    set u($i) [$ns node]
    set m($i) [$ns node]
    $ns duplex-link $u($i) $r1 10Mb 23ms DropTail
    $ns duplex-link $m($i) $r2 10Mb 23ms DropTail
    set tcp3($i) [new Agent/TCP]
    $tcp3($i) set packetSize_ 500
    $ns attach-agent $u($i) $tcp3($i)
    set sink3($i) [new Agent/TCPSink]

```

```

$ns attach-agent $m($i) $sink3($i)
$ns connect $tcp3($i) $sink3($i)
set ftp3($i) [new Application/FTP]
$ftp3($i) attach-agent $tcp3($i)
    $ns at 0.7 "$ftp3($i) start"
    $ns at 49.9 "$ftp3($i) stop"
}
#-----
#TPC4
for {set i 0} {$i < $flows} {incr i} {
    set v($i) [$ns node]
    set n($i) [$ns node]
    $ns duplex-link $v($i) $r1 10Mb 7ms DropTail
    $ns duplex-link $n($i) $r2 10Mb 7ms DropTail
    set tcp4($i) [new Agent/TCP]
    $tcp4($i) set packetSize_ 900
    $ns attach-agent $v($i) $tcp4($i)
    set sink4($i) [new Agent/TCPSink]
    $ns attach-agent $n($i) $sink4($i)
    $ns connect $tcp4($i) $sink4($i)
    set ftp4($i) [new Application/FTP]
    $ftp4($i) attach-agent $tcp4($i)
        $ns at 0.11 "$ftp4($i) start"
        $ns at 49.9 "$ftp4($i) stop"
    }
}
#-----
#TPC5
for {set i 0} {$i < $flows} {incr i} {
    set w($i) [$ns node]
    set o($i) [$ns node]
    $ns duplex-link $w($i) $r1 10Mb 8ms DropTail
    $ns duplex-link $o($i) $r2 10Mb 8ms DropTail
    set tcp5($i) [new Agent/TCP]
    $tcp5($i) set packetSize_ 1200
    $ns attach-agent $w($i) $tcp5($i)
    set sink5($i) [new Agent/TCPSink]
    $ns attach-agent $o($i) $sink5($i)
    $ns connect $tcp5($i) $sink5($i)
    set ftp5($i) [new Application/FTP]
    $ftp5($i) attach-agent $tcp5($i)
        $ns at 0.17 "$ftp5($i) start"
        $ns at 49.9 "$ftp5($i) stop"
    }
}
#-----
#FIM DO PROG
proc finish {} {
    global ns tracefile1 #namfile
    $ns flush-trace
    close $tracefile1
    exit 0
}
#-----
#COMANDOS
$ns at 5.0 "$video1 start"
$ns at 49.0 "$video1 stop"

$ns at 50.0 "$null1 closefile"
$ns at 50.0 "finish"
$ns run

```

Anexo II

Parâmetros de Codificação dos Vídeos

```
# New Input File Format is as follows
# <ParameterName> = <ParameterValue> # Comment
#
# See configfile.h for a list of supported ParameterNames

#####
###
# Files
#####
###
#InputFile          = "news_cif.yuv"          # Input sequence
#InputFile          = "coastguard_cif.yuv"     # Input sequence
InputHeaderLength   = 0          # If the inputfile has a header, state it's length in
byte here
StartFrame          = 0          # Start frame for encoding. (0-N)
FramesToBeEncoded   = 300        # Number of frames to be coded
FrameRate           = 30.0       # Frame Rate per second (0.1-100.0)
SourceWidth         = 352        # Frame width
SourceHeight        = 288        # Frame height
TraceFile           = ""
ReconFile           = ""
#OutputFile         = "news_cif_GOP12.264"
#OutputFile         = "coast_cif_GOP12.264"

#####
###
# Encoder Control
#####
###
ProfileIDC          = 100        # Profile IDC (66=baseline, 77=main, 88=extended; FREXT
Profiles: 100=High, 110=High 10, 122=High 4:2:2, 144=High 4:4:4, for params see below)
LevelIDC            = 40         # Level IDC (e.g. 20 = level 2.0)

IntraPeriod         = 12        # Period of I-Frames (0=only first)
EnableOpenGOP       = 0         # Support for open GOPs (0: disabled, 1: enabled)
IDRIntraEnable      = 1         # Force IDR Intra (0=disable 1=enable)
QPISlice            = 28        # Quant. param for I Slices (0-51)
QPPSslice           = 28        # Quant. param for P Slices (0-51)
FrameSkip           = 0         # Number of frames to be skipped in input (e.g 2 will code
every third frame)
ChromaQPOffset      = 0         # Chroma QP offset (-51..51)
UseHadamard         = 1         # Hadamard transform (0=not used, 1=used for all subpel
positions, 2=use only for qpel)
DisableSubpelME     = 0         # Disable Subpixel Motion Estimation (0=off/default, 1=on)
SearchRange         = 16        # Max search range
NumberReferenceFrames = 5        # Number of previous frames used for inter motion search
(1-16)
PList0References    = 0         # P slice List 0 reference override (0 disable, N <=
NumberReferenceFrames)
Log2MaxFNumMinus4   = 0         # Sets log2_max_frame_num_minus4 (-1 : based on
FramesToBeEncoded/Auto, >=0 : Log2MaxFNumMinus4)
Log2MaxPOCLsbMinus4 = -1        # Sets log2_max_pic_order_cnt_lsb_minus4 (-1 : Auto, >=0 :
Log2MaxPOCLsbMinus4)

GenerateMultiplePPS = 0         # Transmit multiple parameter sets. Currently parameters
basically enable all WP modes (0: disabled, 1: enabled)
ResendPPS           = 0         # Resend PPS (with pic_parameter_set_id 0) for every coded
Frame/Field pair (0: disabled, 1: enabled)
```

```

MbLineIntraUpdate      = 0 # Error robustness(extra intra macro block updates)(0=off,
N: One GOB every N frames are intra coded)
RandomIntraMBRefresh   = 0 # Forced intra MBs per picture
InterSearch16x16        = 1 # Inter block search 16x16 (0=disable, 1=enable)
InterSearch16x8         = 1 # Inter block search 16x8  (0=disable, 1=enable)
InterSearch8x16         = 1 # Inter block search 8x16  (0=disable, 1=enable)
InterSearch8x8          = 1 # Inter block search 8x8   (0=disable, 1=enable)
InterSearch8x4          = 1 # Inter block search 8x4   (0=disable, 1=enable)
InterSearch4x8          = 1 # Inter block search 4x8   (0=disable, 1=enable)
InterSearch4x4          = 1 # Inter block search 4x4   (0=disable, 1=enable)

IntraDisableInterOnly  = 0 # Apply Disabling Intra conditions only to Inter Slices
(0:disable/default,1: enable)
Intra4x4ParDisable      = 0 # Disable Vertical & Horizontal 4x4
Intra4x4DiagDisable     = 0 # Disable Diagonal 45degree 4x4
Intra4x4DirDisable      = 0 # Disable Other Diagonal 4x4
Intra16x16ParDisable    = 0 # Disable Vertical & Horizontal 16x16
Intra16x16PlaneDisable  = 0 # Disable Planar 16x16
ChromaIntraDisable      = 0 # Disable Intra Chroma modes other than DC

DisposableP            = 0 # Enable Disposable P slices in the primary layer (0:
disable/default, 1: enable)
DispPQPOffset          = 0 # Quantizer offset for disposable P slices (0: default)

#####
###
# B Slices
#####
###

NumberBFrames          = 0 # Number of B coded frames inserted (0=not used)
QPBSlice               = 30 # Quant. param for B slices (0-51)
BRefPicQPOffset        = 0 # Quantization offset for reference B coded pictures (-
51..51)
DirectModeType         = 1 # Direct Mode Type (0:Temporal 1:Spatial)
DirectInferenceFlag    = 1 # Direct Inference Flag (0: Disable 1: Enable)
BList0References       = 0 # B slice List 0 reference override (0 disable, N <=
NumberReferenceFrames)
BList1References       = 1 # B slice List 1 reference override (0 disable, N <=
NumberReferenceFrames)
                        # 1 List1 reference is usually recommended for normal GOP
Structures.
                        # A larger value is usually more appropriate if a more
flexible
                        # structure is used (i.e. using PyramidCoding)

BReferencePictures      = 0 # Referenced B coded pictures (0=off, 1=on)

PyramidCoding           = 0 # B pyramid (0= off, 1= 2 layers, 2= 2 full pyramid, 3 =
explicit)
PyramidLevelQPEnable    = 1 # Adjust QP based on Pyramid Level (in increments of 1).
Overrides BRefPicQPOffset behavior.(0=off, 1=on)
ExplicitPyramidFormat   = "b2r28b0e30b1e30b3e30b4e30" # Explicit Enhancement GOP. Format
is {FrameDisplay_orderReferenceQP}.
                        # Valid values for reference type
is r:reference, e:non reference.
PyramidRefReorder       = 1 # Reorder References according to Poc distance for
PyramidCoding (0=off, 1=enable)
PocMemoryManagement    = 1 # Memory management based on Poc Distances for
PyramidCoding (0=off, 1=on)

BiPredMotionEstimation = 0 # Enable Bipredictive based Motion Estimation (0:disabled,
1:enabled)
BiPredMERefinements    = 3 # Bipredictive ME extra refinements (0: single, N: N extra
refinements (1 default)
BiPredMESearchRange     = 16 # Bipredictive ME Search range (8 default). Note that
range is halved for every extra refinement.

```

```

BiPredMESubPel          = 1    # Bipredictive ME Subpixel Consideration (0: disabled, 1:
single level, 2: dual level)

#####
###
# SP Frames
#####
###

SPPicturePeriodicity  = 0    # SP-Picture Periodicity (0=not used)
QPSPSlice              = 28   # Quant. param of SP-Slices for Prediction Error (0-51)
QPSP2Slice             = 27   # Quant. param of SP-Slices for Predicted Blocks (0-51)

#####
###
# Output Control, NALs
#####
###

SymbolMode             = 1    # Symbol mode (Entropy coding method: 0=UVLC, 1=CABAC)
OutFileMode            = 0    # Output file mode, 0:Annex B, 1:RTP
PartitionMode          = 0    # Partition Mode, 0: no DP, 1: 3 Partitions per Slice

#####
###
# CABAC context initialization
#####
###

ContextInitMethod      = 1    # Context init (0: fixed, 1: adaptive)
FixedModelNumber       = 0    # model number for fixed decision for inter slices (
0, 1, or 2 )

#####
###
# Interlace Handling
#####
##

PicInterlace           = 0    # Picture AFF      (0: frame coding, 1: field coding,
2:adaptive frame/field coding)
MbInterlace            = 0    # Macroblock AFF (0: frame coding, 1: field coding,
2:adaptive frame/field coding)
IntraBottom            = 0    # Force Intra Bottom at GOP Period

#####
###
# Weighted Prediction
#####
##

WeightedPrediction      = 0    # P picture Weighted Prediction (0=off, 1=explicit
mode)
WeightedBiprediction    = 0    # B picture Weighted Prediciton (0=off, 1=explicit
mode, 2=implicit mode)
UseWeightedReferenceME  = 0    # Use weighted reference for ME (0=off, 1=on)

#####
###
# Picture based Multi-pass encoding
#####
##

RDPictureDecision      = 0    # Perform RD optimal decision between different coded
picture versions.

```

```

# If GenerateMultiplePPS is enabled then this will
test different WP methods.
# Otherwise it will test QP +-1 (0: disabled, 1:
enabled)
RDPictureIntra          = 0      # Perform RD optimal decision also for intra coded
pictures (0: disabled (default), 1: enabled).
RDPSSliceWeightOnly     = 1      # Only consider Weighted Prediction for P slices in
Picture RD decision. (0: disabled, 1: enabled (default))
RDBSliceWeightOnly      = 0      # Only consider Weighted Prediction for B slices in
Picture RD decision. (0: disabled (default), 1: enabled )

#####
###
# Loop filter parameters
#####
###

LoopFilterParametersFlag = 0      # Configure loop filter (0=parameter below ingored,
1=parameters sent)
LoopFilterDisable        = 0      # Disable loop filter in slice header (0=Filter, 1=No
Filter)
LoopFilterAlphaC0Offset  = 0      # Alpha & C0 offset div. 2, {-6, -5, ... 0, +1, ..
+6}
LoopFilterBetaOffset     = 0      # Beta offset div. 2, {-6, -5, ... 0, +1, .. +6}

#####
###
# Error Resilience / Slices
#####
###

SliceMode                = 0      # Slice mode (0=off 1=fixed #mb in slice 2=fixed #bytes in
slice 3=use callback)
SliceArgument            = 50      # Slice argument (Arguments to modes 1 and 2 above)

num_slice_groups_minus1 = 0      # Number of Slice Groups Minus 1, 0 == no FMO, 1 == two
slice groups, etc.
slice_group_map_type      = 0      # 0: Interleave, 1: Dispersed,      2: Foreground with
left-over,
                                # 3: Box-out,      4: Raster Scan   5: Wipe
                                # 6: Explicit, slice_group_id read from

SliceGroupConfigFileName
slice_group_change_direction_flag = 0      # 0: box-out clockwise, raster scan or wipe
right,
                                # 1: box-out counter clockwise, reverse raster
scan or wipe left
slice_group_change_rate_minus1    = 85      #
SliceGroupConfigFileName          = "sg0conf.cfg"      # Used for slice_group_map_type 0,
2, 6

UseRedundantSlice          = 0      # 0: not used, 1: one redundant slice used for each slice
(other modes not supported yet)

#####
###
# Search Range Restriction / RD Optimization
#####
###

RestrictSearchRange        = 2      # restriction for (0: blocks and ref, 1: ref, 2: no
restrictions)
RDOOptimization            = 1      # rd-optimized mode decision
                                # 0: RD-off (Low complexity mode)
                                # 1: RD-on (High complexity mode)
                                # 2: RD-on (Fast high complexity mode - not work in FREX
Profiles)
                                # 3: with losses

```

```

DisableThresholding = 0 # Disable Thresholding of Transform Coefficients (0:off,
1:on)
DisableBSkipRDO      = 0 # Disable B Skip Mode consideration from RDO Mode decision
(0:off, 1:on)
SkipIntraInInterSlices = 0 # Skips Intra mode checking in inter slices if certain
mode decisions are satisfied (0: off, 1: on)

# Explicit Lambda Usage
UseExplicitLambdaParams = 0 # Use explicit lambda scaling parameters (0:disabled,
1:enabled)
LambdaWeightISlice      = 0.65 # scaling param for I slices. This will be used as a
multiplier i.e. lambda=LambdaWeightISlice * 2^((QP-12)/3)
LambdaWeightPSlice      = 0.68 # scaling param for P slices. This will be used as a
multiplier i.e. lambda=LambdaWeightPSlice * 2^((QP-12)/3)
LambdaWeightBSlice      = 2.00 # scaling param for B slices. This will be used as a
multiplier i.e. lambda=LambdaWeightBSlice * 2^((QP-12)/3)
LambdaWeightRefBSlice   = 1.50 # scaling param for Referenced B slices. This will be
used as a multiplier i.e. lambda=LambdaWeightRefBSlice * 2^((QP-12)/3)
LambdaWeightSPslice     = 1.50 # scaling param for SP slices. This will be used as a
multiplier i.e. lambda=LambdaWeightSPslice * 2^((QP-12)/3)
LambdaWeightSISlice     = 0.65 # scaling param for SI slices. This will be used as a
multiplier i.e. lambda=LambdaWeightSISlice * 2^((QP-12)/3)

LossRateA               = 10 # expected packet loss rate of the channel for the first
partition, only valid if RDOptimization = 2
LossRateB               = 0 # expected packet loss rate of the channel for the second
partition, only valid if RDOptimization = 2
LossRateC               = 0 # expected packet loss rate of the channel for the third
partition, only valid if RDOptimization = 2
NumberOfDecoders        = 30 # Numbers of decoders used to simulate the channel, only
valid if RDOptimization = 2
RestrictRefFrames       = 0 # Doesnt allow reference to areas that have been intra
updated in a later frame.

#####
##
# Additional Stuff
#####
##

UseConstrainedIntraPred = 0 # If 1, Inter pixels are not used for Intra macroblock
prediction.
LastFrameNumber         = 0 # Last frame number that have to be coded (0: no effect)
ChangeQPI               = 16 # QP (I-slices) for second part of sequence (0-51)
ChangeQPP               = 16 # QP (P-slices) for second part of sequence (0-51)
ChangeQPB               = 18 # QP (B-slices) for second part of sequence (0-51)
ChangeQPBsRefOffset     = 2 # QP offset (stored B-slices) for second part of
sequence (-51..51)
ChangeQPStart           = 0 # Frame no. for second part of sequence (0: no second
part)

NumberOfLeakyBuckets    = 8 # Number of Leaky Bucket values
LeakyBucketRateFile     = "leakybucketrate.cfg" # File from which encoder derives
rate values
LeakyBucketParamFile    = "leakybucketparam.cfg" # File where encoder stores
leakybucketparams

NumberFramesInEnhancementLayerSubSequence = 0 # number of frames in the Enhanced
Scalability Layer(0: no Enhanced Layer)
NumberOfFrameInSecondIGOP = 0 # Number of frames to be coded in the
second IGOP

SparePictureOption      = 0 # (0: no spare picture info, 1: spare picture
available)
SparePictureDetectionThr = 6 # Threshold for spare reference pictures detection
SparePicturePercentageThr = 92 # Threshold for the spare macroblock percentage

PicOrderCntType         = 0 # (0: POC mode 0, 1: POC mode 1, 2: POC mode 2)

```

```
#####
#
#Rate control
#####
#

RateControlEnable      =      0      # 0 Disable, 1 Enable
Bitrate                =    128000    # Bitrate(bps)
InitialQP              =      24      # Initial Quantization Parameter for the first I frame
                                # InitialQp depends on two values: Bits Per Picture,
                                # and the GOP length
BasicUnit              =      11      # Number of MBs in the basic unit
                                # should be a fractor of the total number
                                # of MBs in a frame
ChannelType            =      0      # type of channel( 1=time varying channel; 0=Constant
channel)

#####
#
#Fast Mode Decision
#####
#
EarlySkipEnable        =      0      # Early skip detection (0: Disable 1: Enable)
SelectiveIntraEnable   =      0      # Selective Intra mode decision (0: Disable 1: Enable)

#####
#
#FREXT stuff
#####
#

YUVFormat              = 1          # YUV format (0=4:0:0, 1=4:2:0, 2=4:2:2, 3=4:4:4)
RGBInput               = 0          # 1=RGB input, 0=GBR or YUV input
BitDepthLuma           = 8          # Bit Depth for Luminance (8...12 bits)
BitDepthChroma         = 8          # Bit Depth for Chrominance (8...12 bits)
CbQPOffset             = 0          # Chroma QP offset for Cb-part (-51..51)
CrQPOffset             = 0          # Chroma QP offset for Cr-part (-51..51)
Transform8x8Mode       = 1          # (0: only 4x4 transform, 1: allow using 8x8 transform
additionally, 2: only 8x8 transform)
ResidueTransformFlag   = 0          # (0: no residue color transform 1: apply residue color
transform)
ReportFrameStats       = 0          # (0:Disable Frame Statistics 1: Enable)
DisplayEncParams       = 0          # (0:Disable Display of Encoder Params 1: Enable)
Verbose               = 1          # level of display verboseness (0:short, 1:normal,
2:detailed)

#####
#
#Q-Matrix (FREXT)
#####
#
QmatrixFile            = "q_matrix.cfg"

ScalingMatrixPresentFlag = 0      # Enable Q_Matrix (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
ScalingListPresentFlag0 = 3      # Intra4x4_Luma (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
ScalingListPresentFlag1 = 3      # Intra4x4_ChromaU (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
ScalingListPresentFlag2 = 3      # Intra4x4_chromaV (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
ScalingListPresentFlag3 = 3      # Inter4x4_Luma (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
ScalingListPresentFlag4 = 3      # Inter4x4_ChromaU (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
ScalingListPresentFlag5 = 3      # Inter4x4_ChromaV (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
```



```

ScalingListPresentFlag6 = 3      # Intra8x8_Luma      (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)
ScalingListPresentFlag7 = 3      # Inter8x8_Luma      (0 Not present, 1 Present in SPS, 2
Present in PPS, 3 Present in both SPS & PPS)

#####
#
#Rounding Offset control
#####
#

OffsetMatrixPresentFlag = 0      # Enable Explicit Offset Quantization Matrices (0:
disable 1: enable)
QOffsetMatrixFile       = "q_offset.cfg" # Explicit Quantization Matrices file

AdaptiveRounding
disable, 1: enable)      = 0      # Enable Adaptive Rounding based on JVT-N011 (0:
AdaptRndPeriod          = 1      # Period in terms of MBs for updating rounding offsets.
                                # 0 performs update at the picture level. Default is
                                # 16. 1 is as in JVT-N011.
AdaptRndChroma          = 0      # Enables coefficient rounding adaptation for chroma

AdaptRndWFactorIRef     = 4      # Adaptive Rounding Weight for I/SI slices in reference
pictures /4096
AdaptRndWFactorPRef     = 4      # Adaptive Rounding Weight for P/SP slices in reference
pictures /4096
AdaptRndWFactorBRef     = 4      # Adaptive Rounding Weight for B slices in reference
pictures /4096
AdaptRndWFactorINRef    = 4      # Adaptive Rounding Weight for I/SI slices in non
reference pictures /4096
AdaptRndWFactorPNRef    = 4      # Adaptive Rounding Weight for P/SP slices in non
reference pictures /4096
AdaptRndWFactorBNRef    = 4      # Adaptive Rounding Weight for B slices in non
reference pictures /4096

#####
#
#Lossless Coding (FREXT)
#####
#

QPPrimeYZeroTransformBypassFlag = 0      # Enable lossless coding when qpprime_y is zero
(0 Disabled, 1 Enabled)

#####
#
#Fast Motion Estimation Control Parameters
#####
#

UseFME                  = 0      # Use fast motion estimation (0=disable/default,
1=UMHexagonS,
                                # 2=Simplified UMHexagonS, 3=EPZS patterns)

EPZSPattern             = 2      # Select EPZS primary refinement pattern.
                                # (0: small diamond, 1: square, 2: extended
                                # diamond/default,
                                # 3: large diamond)
EPZSDualRefinement      = 3      # Enables secondary refinement pattern.
                                # (0:disabled, 1: small diamond, 2: square,
                                # 3: extended diamond/default, 4: large diamond)
EPZSFixedPredictors     = 2      # Enables Window based predictors
                                # (0:disabled, 1: P only, 2: P and B/default)
EPZSTemporal            = 1      # Enables temporal predictors
                                # (0: disabled, 1: enabled/default)
EPZSSpatialMem          = 1      # Enables spatial memory predictors
                                # (0: disabled, 1: enabled/default)
EPZSMinThresScale       = 0      # Scaler for EPZS minimum threshold (0 default).

```

```
EPZSMedThresScale    = 1    # Increasing value can speed up encoding.  
                        # Scaler for EPZS median threshold (1 default).  
EPZSMaxThresScale    = 1    # Increasing value can speed up encoding.  
                        # Scaler for EPZS maximum threshold (1 default).  
                        # Increasing value can speed up encoding.
```

Anexo III

Artigo publicado no CITTEL XIV CCIA 08

Daniel P. Torres, Evelio M. G. Fernandez, Vitalio A. Reguera, Cezar de Oliveira, "On the Impact of Active Queue Management in IP Networks Transporting H.264/MPEG-4 AVC Video Streams", CITTEL XIV CCIA 08, 1 al 5 de Diciembre, 2008, Habana, Cuba. ISBN: 978-959-261-281-5.

On the Impact of Active Queue Management in IP Networks Transporting H.264/MPEG-4 AVC Video Streams

Daniel P. Torres, Evelio M. G. Fernandez, Vitalio A. Reguera, Cezar de Oliveira

Abstract — Considering the adoption of the H.264/MPEG-4 AVC video codec as the base-standard for the latest services and products for streaming and storage of digital video, this work assesses the impact on the QoS offered by IP networks that transport real-time H.264/MPEG-4 AVC video streams when applying ARED as an active queue management technique replacing the traditional drop-tail technique. The results obtained show a substantial qualitative gain in the transmission of video streams with the adoption of the ARED technique in the routers, both in terms of queuing delay and jitter, and in terms of video packets loss.

Key Words — Active queue management; H.264/MPEG-4 AVC over IP; multimedia streams; quality of service (QoS).

I INTRODUCTION

With the growing popularity of real-time multimedia transmissions over the Internet, monitoring the quality perceived by the users of such services and assessing the influence that multimedia streams have over the other types of streams that exist in the Internet have become crucial.

Multimedia applications are getting more and more common, consuming an ever growing share of the resources of the world wide web. Two main reasons explain this growth. The first one is the dissemination of broad-band accesses. Real-time video streams require large bandwidth, so access networks with larger transmission capabilities naturally stimulate the transmission of large amounts of information like digital video streams. The second reason is the improvement level that the technology of voice, audio and video compression reached in the last few years.

In the particular case of video compression, the most recent and significant improvement is the H.264/MPEG-4 AVC (Advanced Video Coding) standard, a joint development of the international group of studies VCEG (Video Coding Experts Group), from ITU-T, and MPEG (Moving Picture Experts Group), from ISO/IEC (International Organization for Standardization / International Electrotechnical Commission), adopted in the most modern services and products for the

transmission and storage of digital video. Reference [1] points out that it would be possible to offer DVD-quality video contents over currently available DSL infrastructure.

Considering the growing importance that H.264/MPEG-4 AVC streams have in packet switched wireline networks that use the Internet Protocol (IP) for data transport, as well as the high variability of the traffic generated by this codec, this work assesses the gain in terms of Quality of Service (QoS) of H.264/MPEG-4 AVC streams when the Active Queue Management (AQM) technique called Adaptive Random Early Detection (ARED) is applied. The results are then compared to the ones obtained through the application of the traditional drop-tail technique. The proposed topology was simulated in the Network Simulator 2 (NS2) software and the parameters of QoS measured were loss of packets, delay and jitter.

Below we discuss the essential aspects involved in the subject of this paper.

A. Transmission of multimedia streams over the Internet

In the current service model used in the Internet, called best effort, implemented through the IP protocol in the network layer, no guaranties can be made to the users regarding the delivery of the information packets. For this reason, as a complement to IP, there are two main protocols implemented in the transport layer that make the end-to-end transport of the packets: TCP (Transmission Control Protocol) e UDP (User Datagram Protocol).

Since the end of the 80s, TCP has been used to transport the majority of Internet content, like e-mail and Web traffic, obeying the so called TCP traffic paradigm [2]. In this paradigm, the loss of packets that causes the reduction of the transmission rate is normally caused by the discard of packets by congested routers that have their queuing buffers overfilled. This discard technique is called drop-tail.

Nevertheless, when the best-effort characteristic of IP and the reliability mechanisms of TCP are combined, the

usage of TCP/IP for the transmission of real-time streams is not suitable. Since a large part of the multimedia applications running over the Internet, such as VoIP, real-time audio and video streams, as well as video-conferences, have very strong conversational characteristics and restrictions in terms of transmission times and bandwidth, another combination of protocols became necessary.

That leads to the combination of UDP and IP, where the good behavior and reliability of TCP are replaced by the simplicity of UDP, which only sends the messages with a minimum of mechanisms involved, with no guarantee of delivery neither protection over duplicated packet arrivals. Basically what UDP does is, in the transmitter side, it passes the information segments on to IP so that they can be routed to its destination and, in the receptor side, the IP segments are passed on to the application [3]. Nevertheless, UDP traffic does not respond to packet losses in the same way that TCP does. In reality, there is no response at all, what provokes an unfair bandwidth division between TCP and UDP traffics.

But UDP was not originally developed for the transport of real-time voice, audio and video through the Internet, it is just the best choice after TCP. For UDP to be used in the transmission of real-time multimedia contents, another protocol is necessary. With that objective in mind, RTP (Real-Time Transport Protocol) was developed, offering end-to-end delivery services for data with real-time characteristics. These services include the identification of the type of data being transported, sequence number, time markers and delivery monitoring. Applications usually run RTP on top of UDP in order to use its multiplexing services, error detection capabilities and packets re-ordering [4].

In [5] there is an extensive study that points out native RTP, together with UDP/IP, as the best choice to transport real-time H.264/MPEG-4 AVC streams.

B. H.264 / MPEG-4 AVC standard overview

Reaching compression rates 50% better than MPEG-2 standard and 30% better than MPEG-4 Part 2 for the same level of perceived quality, the H.264/MPEG-4 AVC standard adds up the most modern techniques of video coding and decoding. The dramatic increase in the compression rate has made H.264/MPEG-4 AVC the standard elected for the majority of the new products and services for video transmission and storage, like video services over DSL lines, video broadcasting to wireless mobile devices, Internet video players and BluRay and HD-DVD optical disks standards [6].

The coding chain of former standards, consisting of block transformation in conjunction with motion compensation and prediction, still exists, but a series of new coding mechanisms were added, providing for a superior performance over previous standards. Furthermore, the H.264/MPEG-4 AVC standard defines several profiles according to the characteristics of both

application and platform, ranging from small processing power platforms in environments with high losses (baseline profile), up to professional applications that only support intra-frame coding, for example see reference [7].

The H.264/MPEG-4 AVC standard defines two layers: the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL). The VCL consists of the central compression mechanism, where the syntactic levels commonly known as block, macro-block and slice are found. It was developed to be as independent of the network as possible. On the other hand, the NAL abstracts and adapts the data generated by the VCL so as to adequate them to the transport through a great variety of communication channels, as well as storage media. Looking to facilitate the integration with the communication channel, the NAL specifies both the byte stream and packet stream formats. The packet stream format is used in applications based on the combined use of the protocols RTP/UDP/IP [7].

C. Constant bit rate versus variable bit rate

There are two main ways of coding video signals: with fixed quantization scales, which result in almost constant quality at the expense of variable bit rate; and with rate control, which adapts the quantization scales so as to keep the transmission rate almost constant, at the expense of variable quality.

This work focuses on video coding using a fixed quantization scale in order to examine the fundamental traffic characteristics of the H.264/MPEG-4 AVC standard [8,9].

D. Traffic variability

In [8] it is shown that the traffic variability generated by the H.264/MPEG-4 AVC standard is significantly higher than the traffic variability generated by the MPEG-2 and MPEG-4 Part 2 standards. While the frame size variation coefficient (defined as the standard deviation divided by the average) reaches levels above 2.4 in the case of H.264/MPEG-4 AVC, the same coefficient is not higher than 1.5 in the cases of MPEG-4 Part 2. Such a coefficient level above 1.5 is unprecedented.

In [10] it is shown that the higher traffic variability generated by the H.264/MPEG-4 AVC coder significantly increases the frame losses if compared to the MPEG-4 Part 2 standard in the transmission of video streams over a bottleneck link.

Such conclusion led us to consider the application of AQM techniques in order to evaluate a possible minimization of frame losses, if compared to the traditional drop-tail technique.

E. Congestion control techniques

The objective of AQM is to reduce the size of the queues and their oscillations, maintaining a high level of link utilization through a fair allocation of resources [11]. AQM techniques that are more efficient and fair than

drop-tail in the congestion control of IP networks have been proposed by several authors and the results regarding losses, delay and jitter are promising.

In [11] there is an extensive study on the impact of AQM in the QoS of VoIP applications. One of the more representative AQM algorithms is analyzed through computational simulation, assessing its effect on the perceived voice quality. The degradation of the transmission is linked to the users' perception through well known algorithmic models, expressing their satisfaction in the MOS (Mean Opinion Score) scale. The main results show that the AQM mechanisms, such as ARED significantly improve the perceived voice quality. This study is enlarged in [12], incorporating other congestion control mechanisms such as IP, REM and AVQ.

In the present work we gave special attention to the application of ARED technique in order to assess the QoS of H.264/MPEG-4 AVC for the following reasons: (i) this technique showed an expressive qualitative gain in the transmission of VoIP streams, which, like real-time video streams, have serious restrictions in terms of timing and loss of information packets [11, 12]; (ii) there is a code already implemented and tested in the simulation software used in this work.

II MATERIALS AND METHODS

The methodology chosen for this work was the computational simulation of an IP network that transmits real-time H.264/MPEG-4 AVC video streams, disputing network resources with other types of traffic commonly found in the Internet.

The simulations were implemented in the Network Simulator version 2 (NS2) and the data analysis was performed off-line from the software's trace files.

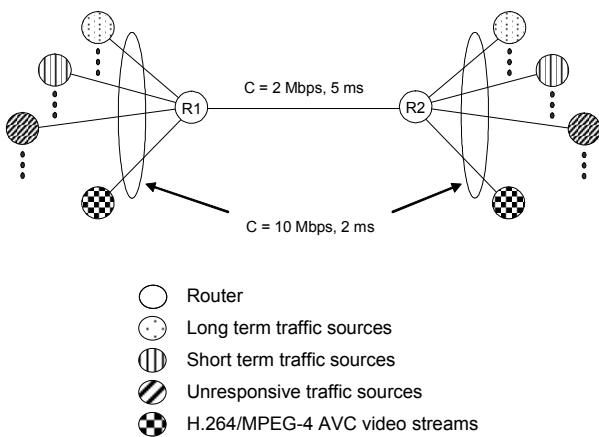


Fig. 1 Simulated network topology.

A. Simulated network topology and configuration

Over a dumbbell topology, where two routers create the main link, secondary links were created between the routers and the traffic sources (Fig. 1). The secondary

links were configured with a 10 Mbps rate and transmission time of 2 ms, while the main link was configured with a 2 Mbps rate and transmission time of 5 ms, characterizing the transmission bottleneck. This topology was selected because it has been traditionally used to evaluate the performance of AQM techniques [13, 14].

Over the secondary links, two traffic configurations were created (see below), and both were simulated with drop-tail and ARED techniques, for further comparison of the results.

Traffic configuration #1: long term (ex.: FTP) and short term traffic (ex. web) were created in the same direction of the video traffic, as well as reverse path traffic. The TCP connections simulating long term traffic was started in the beginning of the simulation, as well as the web and reverse path traffics. The video transmission was started five seconds later. The TCP segments were fixed in 1000 bytes, which is the average size of the segments used in the Internet. Unresponsive streams were also added using UDP agents, simulating bidirectional VoIP streams.

Under such conditions, we measured the delay and jitter of the video packets transmitted, where each simulation comprehended five successive repetitions (or a total of 50 seconds) of the video streams news_cif and coastguard_cif described below.

Traffic configuration #2: only long term traffic was created, using TCP connections with the same average size of segments as above, but the number of simultaneous connections was varied between 0 and 30.

Under such conditions, we measured packet and frame losses of the video streams news_cif and coastguard_cif, as well as the effective information exchange rate between source and destination of TCP connections (goodput).

In both configurations, the AQM technique ARED was implemented with queues of 120 packets. The parameters were set according to the general guidelines suggested in [13]. The target queue size was set to 20 packets which corresponds to an average queuing delay of 40 ms considering an average packet size of 500 bytes.

B. Simulated video streams

The H.264/MPEG-4 AVC video streams were generated from raw video sequences in the CIF 352x288 pixels, 4:2:0 YUV format, originally sampled at 30 frames per second. In this work we used the sequences news_cif.yuv and coastguard_cif.yuv, both available at <http://trace.eas.asu.edu/yuv/index.html>.

These two video sequences were chosen because they present rather diverse characteristics, both in terms of compression ratio and traffic variability. While coastguard_cif presents a high rate of image variation between frames, the sequence news_cif presents images with low variation rate. The result is high compression ratio and high traffic variability in the case of the video sequence news_cif, and low compression ratio and low

traffic variability in the case of the video sequence `coastguard_cif`. In table I we list the most relevant statistical data of the encoded video sequences.

TABLE I
STATISTICAL DATA OF THE ENCODED VIDEO SEQUENCES.

	news_cif	coastguard_cif
Number of frames	300	300
Compression ratio	110,54	33,42
Average frame size (bytes)	1392,54	4614,04
Minimum frame size (bytes)	70	2219
Maximum frame size (bytes)	8603	14581
Peak-to-mean ratio	6,18	3,16
Total size of the sequence (bytes)	417761	1384214
Average transmission rate (bits/s)	334208,8	1107371,2

Encoding was done through JM reference software version 13.0, using “High” profile. The GOP (group of pictures) used was one intra-coded frame (type I) for every 11 inter-coded frames (type P), with no bi-directional inter-coded frames (type B), resulting in a closed GOP IPPPPPPPPPP. The quantization parameter was set to 28, and each frame was segmented in packets 1024 bytes long.

Flow control was disabled for two reasons: i) coding was done offline; ii) the objective of the simulation is to assess the behavior of the other concurrent streams interacting with a high variability stream. So the output of the encoder resulted in a video stream of unconstrained variable bit rate (VBR), without any interaction between the encoder and network conditions.

Through the MP4TRACE tool, the encoded video sequence was used as the input for the generation of a text-type “trace” file, which contains all the necessary information for NS2 to create a simulated stream that behaves like the original encoded stream in terms of frame sequence number, type of frame, quantity and size of frame segments (packets) sent, and transmission timing [15]. The tools that were used are described in [16, 17].

III RESULTS AND DISCUSSION

The application of ARED technique showed significant qualitative gains if compared to drop-tail technique in all measured parameters, as detailed below:

A. Delay and jitter

The results indicate that the ARED technique is able to keep the average delay of the video packets within the configured queuing delay (40 ms) even under severe congestion conditions (traffic configuration #1). Besides that, measured jitter was significantly lower if compared to the drop-tail technique. Such results lead to the following perceptions: (i) the losses caused by excessively late arrivals can be minimized; (ii) the size of the jitter buffer needed to absorb the delay variations can also be minimized.

Figures 2 and 3 show the delay measurements of the

video sequence `coastguard_cif` after five successive repetitions.

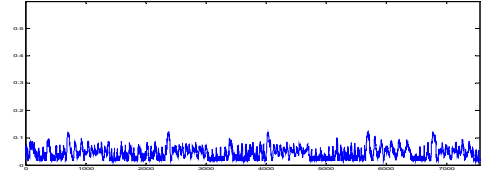


Fig. 2 Video packets delay – ARED technique.

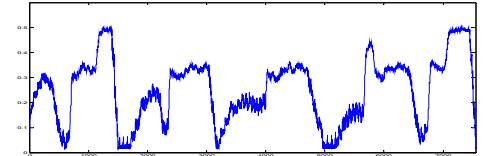


Fig. 3 Video packets delay – drop-tail technique.

B. Packet losses and video information losses

Packet losses and video information losses due exclusively to discards at the routers were significantly reduced with the application of ARED technique, even under severe congestion conditions, when compared to the measurements done under drop-tail regime, as shown in Figures 4 and 5. Since the loss of information packets is the main degradation caused in digital video streams transmitted in IP networks, reductions of 40% to 60%, as shown in the graphs below, possibly indicate an expressive improvement in the QoS perceived by the user, besides representing an ease on the burden of the application in order to minimize the effects of image quality degradation through the use of error concealment techniques.

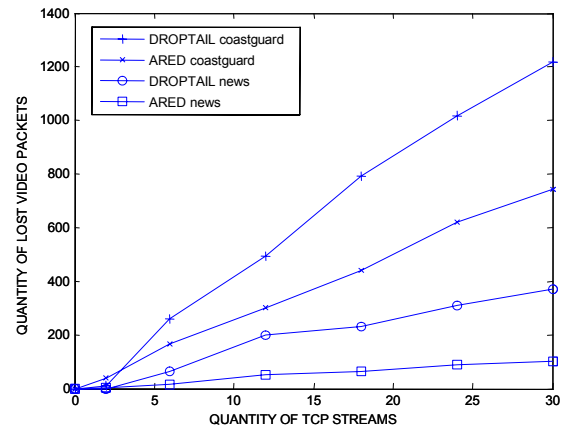


Fig. 4 Video packets losses: quantity of lost video packets versus quantity of TCP streams.

In order to measure the impact of the ARED technique also on the concurrent TCP streams, their average goodput was assessed under the multiple simulated load conditions. According to Figure 6, the goodput of the TCP connections was kept almost constant, even under heavy traffic conditions. The small reduction of the

goodput with the application of the ARED technique can be considered insignificant if compared to the reduction of the video packet losses.

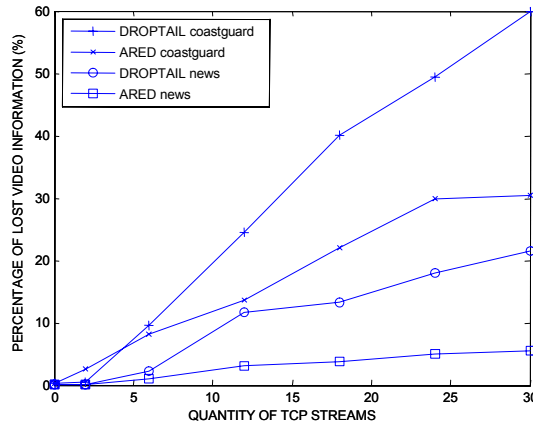


Fig. 5 Video information losses (%): percentage of video information lost versus quantity of TCP streams.

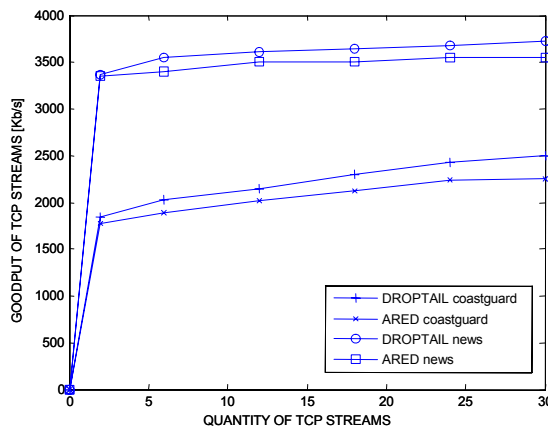


Fig. 6 TCP connections goodput.

IV CONCLUSION

The results above show that H.264/MPEG-4 AVC video streams greatly benefit from the ARED active queue management technique implemented in the routers that form the bottleneck transmission link of the simulated network. Despite the high variability of the traffic generated by the encoder, the average delay and jitter were drastically reduced if compared to the drop-tail technique, what is reflected as lower losses due to excessive delays, besides the reduction of the size of the jitter buffer needed to absorb the delay variations. An expressive reduction in the losses of video information caused exclusively by the discards at the routers was perceived, without significantly degrading the concurrent TCP connections. The observations above indicate a great improvement in the QoS perceived by the user, but this observation needs to be confirmed through specific metrics. So, as a suggestion for future work is the evaluation of the QoS through parameters that possess a better correlation with the final user's perception, like the PSNR – Peak Signal to Noise Ratio measured frame by frame, with a corresponding mapping to a MOS scale.

Also as a suggestion for future work is the implementation of other queuing techniques like PI, REM and AVQ, for a broader comparative study.

ACKNOWLEDGMENT

Daniel Pioli Torres would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for granting the Masters Degree scholarship that allowed the development of this research.

REFERENCES

- [1] White paper - Intel, "H.264 & IPTV Over DSL – Enabling New Telco Revenue Opportunities", 2004.
- [2] S. Wenger, "H.264/AVC Over IP", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, 2003.
- [3] L. Soares, "Redes de Computadores: das LANs, MANs e WANs às Redes ATM", Ed. Campus, 1995.
- [4] M. Santos, "Medidas de Qualidade de Voz em Redes IP", Masters Degree Dissertation, UFPR, 2006.
- [5] White paper – Envivio, "IP Streaming of MPEG-4: Native RTP vs MPEG-2 Transport Stream", October 2005.
- [6] White paper – The MPEG Industry Forum, "Understanding MPEG-4: Technologies, Advantages, and Markets", 2005.
- [7] S. Kwon, A. Tamhankar, K. Rao, "Overview of H.264/MPEG-4 Part 10", Journal of Visual Communication and Image Representation, vol. 17, 2006.
- [8] G. Auwera, P. David, M. Reisslein, "Traffic Characteristics of H.264/AVC Variable Bit Rate Video", accepted in IEEE Communications Magazine, 2008.
- [9] T. Lakshman, A. Ortega, A. Reibman, "VBR video: tradeoffs and potentials", Proceedings of the IEEE, vol. 86, no. 5, pp. 952-973, May 1998.
- [10] G. Auwera, P. David, M. Reisslein, "Traffic and Quality Characterization of Single-Layer Video Streams Encoded with the H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension", accepted in IEEE Transactions on Broadcasting: Quality Issues in Multimedia Broadcasting, 2008.
- [11] V. Reguera, F. Paliza, E. Fernández, and W. Godoy Jr., "Voice over IP Quality of Service Using Active Queue Management", Proceedings of the VI International Telecommunications Symposium (ITS2006), Fortaleza-CE, Brasil, September 2006.
- [12] V. Reguera, F. Paliza, E. Fernández, and W. Godoy Jr., "On the Impact of Active Queue Management on Voice over IP Quality of Service", Computer Communications, Vol. 31, No.1, pp. 73-87, January 2008.
- [13] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", IEEE/ACM Trans. Networking, vol. 1, no. 4, August 1993, pp. 397-413.
- [14] S. Floyd, R. Gummadi, S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management", Available online: <http://www.icir.org/floyd/papers/adaptiveRed.pdf>, August 1, 2001.
- [15] A. Ziviani, J. Rezende, O. Duarte, S. Fdida, "Improving the Delivery Quality of MPEG Video Streams by Using Differentiated Services", IEEE Potentials, January/February 2007, pp. 16-19.
- [16] J. Klaue, B. Rathke, A. Wolisz, "EvalVid – A Framework for Video Transmission and Quality Evaluation", 13th International Conference on Modeling Techniques and Tools for Computer Evaluation, Urbana, Illinois, USA, 2003.
- [17] Chih-Heng Ke, Ce-Kuen Shieh, Wen-Shyang Hwang, Artur Ziviani, "An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission", Journal of Information Science and Engineering (JISE), Vol: 24; No: 2; Pág: 425-440; 2008.

